The Application of Multi-Agent Systems to the Design of an Intelligent Geometry Compressor

A thesis submitted for the degree of Doctor of Philosophy

by

Gwyn Morgan

Department of Information Systems and Computing
BRUNEL UNIVERSITY

July 2002

ABSTRACT

In this research, a multi-agent approach was applied to the design of a large axial flow compressor in order to optimise performance and to greatly enlarge the useful operating range of the machine. In this design a number of distributed software/hardware agents co-operate to control the internal geometry of the machine and thereby optimise the compressor characteristics in response to changes in flow conditions. The resulting machine is termed an ‘Intelligent Geometry Compressor’ (IGC).

The design of a multi-agent system for the IGC was carried out in three main phases, each supported by computer simulation. In the first phase a steady-state model of the IGC was developed in which global control of the variable geometry is achieved by a single agent. This was used to help identify specific requirements for performance and the underlying parametric relationships. The subsequent phases incorporated additional agents into the machine design to meet these requirements. Initially, agents were deployed to optimise the settings of individual rows of stator vanes. In the final phase, the MAS was extended to incorporate agents into the machine design for the control of individual stator vanes.

Simulation results were obtained which demonstrate the effectiveness of the intelligent geometry compressor in achieving delivery pressure regulation over a wide range of steady-state operating conditions whilst optimising overall machine efficiency and avoiding the occurrence of stall. Some of the implications for the physical design of an IGC arising from the MAS concept were briefly considered.

The experience of the research supported by the specific results and observations from many simulation trials, led to the conclusion that multi-agent systems can provide an effective and novel alternative approach to the design of an intelligent geometry compressor. By implication, this conclusion may be extended to other intelligent machine applications where similar opportunity to apply a distributed control solution exists.

Acknowledgements

Sincere thanks are owed to Professor George Rzevski who provided the inspiration for this research and who has also served as my supervisor during the past four and a half years. I am grateful for the many hours of personal time and attention which George has given to me and for his timely and expert guidance. I would also like to express my gratitude to the Department of Information Systems and Computing of Brunel University for their support and sponsorship of the work. Finally, I must acknowledge the support and forebearance of my wife and family without which this largely 'home-based' research would not have been possible.

Gwyn Morgan
Stroud
July 2002

CONTENTS

CONTENTS (continued)

ILLUSTRATIONS

ILLUSTRATIONS (continued)

ILLUSTRATIONS (continued)

CHAPTER 1

Introduction

Research Aims

The specific aim of this research is to demonstrate how a multi-agent systems (MAS) approach may be applied to the design of an axial flow compressor in order to optimise performance and considerably enlarge the useful operating envelope of the machine. This application was proposed by Rzevski (1998) as one of a number of potential engineering applications for multi-agent systems and termed an 'Intelligent Geometry Compressor' (IGC). The underlying hypothesis of the research is that multi-agent systems offer a novel, and potentially effective, alternative approach to the design of engineering systems and machines which are required to operate in a largely autonomous and intelligent manner.

Background

Machines provide solutions to specific engineering problems found in a wide range of industrial applications. Here, the meaning of the term *machine* is a device or piece of equipment, comprising mechanical, electronic and software components, which carries out some significant, purposeful function typically involving the transmission or conversion of energy. Examples of machines in industry include robotic manipulators, automatic guided vehicles, machine tools, assembly and test equipment, power plant such as turbines and compressors, material handling and process control equipment.

A general consequence of the continuing growth in capability per unit cost of computer, communications and information technology is the potential for enhancing machine functionality and control. This, in turn, offers opportunity to extend and optimise the performance of existing machines or to design totally new machines for new applications which hitherto would have been impractical or prohibitively expensive. Technical advantages sought include increased operating range, greater energy efficiency, higher precision, faster response, greater durability and reduced physical size whilst economic motivation lies in reduced lifetime costs of machines and increased productivity. However, this enlargement of machine performance demands a more complex, adaptable and autonomous behaviour than that previously achieved. Machines that exhibit this capability are invariably described as 'intelligent'.

The concept of an *intelligent machine* has its origin in the field of artificial intelligence (O'Hare and Jennings, 1996) associated with the design of expert computer systems. The adoption of the term for engineering applications has followed and references to intelligent machines appear in works on Mechatronics (Rzevski, 1994), and Intelligent Control Systems (Gupta and Sinha, 1996). Definitions of machine intelligence given in the literature vary depending on context but there appears broad agreement that intelligence is linked with an ability to cope successfully with complexity and uncertainty in the machine operating conditions, the latter arising from unpredictable events or deficient information about those events (Rzevski 1994, Antsaklis and Passino 1993, Lu 1996). It may be noted that this emphasis on uncertainty accords with the remark made by Jean Piaget about human intelligence that "Intelligence is what you use when you don't know what to do." (Calvin, 1996).

By definition, uncertainty in the operating conditions of a machine cannot be anticipated fully by the designer of an intelligent machine in terms, for example, of explicit pre-programmed rules. Instead, the machine must embody some means of adapting its behaviour in response to unpredictable events. In principle, there appears to be two basic ways in which this may be achieved. Firstly, the machine may have a capability for learning from past experience and is able to modify existing rules or synthesise new ones to cope with changing operating conditions. Techniques based on Artificial Intelligence concepts, such as neural networks and fuzzy logic, for example, have been developed to support this type of approach (Jang and Sun, 1995). Alternatively, the machine may have the potential for a very large number of behavioural modes arising from the interaction of many distributed, i.e. parallel, processes such that an effective response to new operating conditions is likely to be achieved. Solutions based on some combination of these two principles are also possible.

Multi-Agent Systems

In multi-agent system design the application is perceived as a set of distributed, connected autonomous agents each able to solve a particular problem or sub-problem in pursuit of a specific local goal. Overall system objectives are realised through the co-operation of agents such that the system exhibits what Ferber (1999) identifies as *collective intelligence*. Such systems have been successfully applied to a number of problems in supply chain management, manufacturing planning and electronic commerce (Jennings and Wooldridge 1998). However, with some notable exceptions such as the Archon project (Wittig 1992), there appears to be little evidence of any widespread use of MAS in the design of intelligent machines for industrial application of the type referred to above. Instead, the design of intelligent machines, or rather their control systems, has usually been based on the concepts and methodologies of Intelligent Control (Antsaklis and Passino 1993). Typically, this approach results in a hierarchical organisation of controllers, some of which may employ, for example, sophisticated knowledge-based and learning strategies to effect intelligent behaviour. A hierarchical organisation, however, means that the control regime is, in effect, centralised.

Whilst centralised control systems have proven successful in many instances, they become increasingly difficult to apply effectively in particularly complex and, or, naturally distributed applications. The main reason for this is that a centralised system ultimately imposes restrictions on the flow and processing of information to and from multiple external sources such as actuators and sensors and can thus become a "bottleneck" to machine performance. Introducing sub-systems to alleviate this inherent weakness adds to the complexity of the machine or system design.

Multi-agent systems, being based on distributed architectures, avoid the problems of centralised control and therefore appear to offer advantage in the design of many types of intelligent machine application. Indeed, a number of researchers including Rzevski (1994), Jennings (1994), Wittig (1992), Khosla and Dillon (1997) have identified multi-agent systems as the ‘new paradigm’ for the design of intelligent systems.

Intelligent Geometry Compressor

A multi-stage axial compressor is used in a wide range of industrial applications where extremely large flow rates of gas at moderate pressure levels are required. Flow rates in excess of 100,000 m^3/hour and pressures up to 7 bara (bar absolute) are typical. (Bloch, 1996). These are necessarily physically large machines, which often have to cope with a potentially wide range of operating conditions due to variation in the properties of the gas being compressed and variation in the demands of the downstream system being supplied. The internal aerodynamic design of axial compressors, however, severely limits the reliable and efficient operating range of the machine. Research and development over many years has demonstrated the potential for enhancing axial compressor performance by the introduction of variable geometry guide vanes (stators) that form part of the internal structure of the machine (Riess and Blöcker, 1987). However, variable geometry poses a significant control problem as well as having major implications for the physical design of the compressor.

In this research, the variable geometry elements of an axial compressor, being spatially distributed throughout the machine, are recognised as forming a naturally distributed application to which a multi-agent system would seem to offer an appropriate and potentially effective solution. In this way, an intelligent machine is conceived, which automatically configures its internal geometry to suit the prevailing operating conditions and thereby achieves a considerably larger operating range than its fixed geometry counterpart.

Research Achievements

For the purposes of this research, attention was focused on the steady-state performance of a hypothetical, 5-stage axial compressor. This provided a specific target for MAS application, albeit one whose value is necessarily limited due to the omission of dynamic behaviour considerations. The main part of the research was the design of a multi-agent system for controlling the hypothetical machine and was carried out in three phases of work. In the first phase, a simple system design was investigated in which a single agent exercised global control of variable stators in all stages of the target machine. In the second phase, a system of 7 agents was introduced which enabled independent control of each variable stator row to be achieved. In the final phase, the MAS was extended to allow independent control of each individual vane within stator rows. This system comprised 106 agents. In both of the resulting multi-agent systems, the agents are reactive, employing largely heuristic control methods, and are organised in a network architecture.

Each phase of MAS design was carried out in the context of an evolving computer simulation program. The program was developed in C++ and simulates the concurrent operation of agents and the resulting overall compressor performance. For purposes of simulation, a mathematical model of the target compressor was created based on a stage prediction method proposed by Howell and Bonham (1950). The use of simulation, as described, represents a significant achievement of the research since it formed such a crucial part of the overall design process, enabling the parametric relationships of compressor flow to be captured and providing insight, through many trials, for MAS development. The resulting agent methods and co-operation strategies, as implemented in the simulation program code, are also significant outcomes of the research and demonstrate the potential of an MAS approach in this application.

Research Report

The first chapter of the report presents an introduction to axial flow compressors and highlights the inherent complexities and difficulties in controlling this type of machine. In the following chapter a detailed account of multi-agent systems is given based on a review of the relevant published literature. This established the theoretical basis for the later design work and also provided opportunity to consider MAS with respect to related fields such as Intelligent Control Systems and Mechatronics. The next two chapters briefly describe, respectively, the methodology adopted for the research and some preliminary considerations of the simulation program.

The main body of the report, comprising Chapters 6, 7 and 8 describes the three phases of design work in detail and presents the results of simulation trials. The implications of the MAS approach for the physical design of an intelligent geometry compressor is then briefly considered in Chapter 9.

Finally, conclusions drawn from the research are presented in Chapter 10 where the achievements of the work are reviewed and discussed with regard to the research aims. In this chapter, also, possible areas for further work are suggested. A detailed analysis of compressor flow, on which the hypothetical target machine was based, is given in the Appendix to the report.

The three versions of simulation program and the related C++ source code are included on a CD-ROM which accompanies this thesis.

CHAPTER 2

Axial Flow Compressor

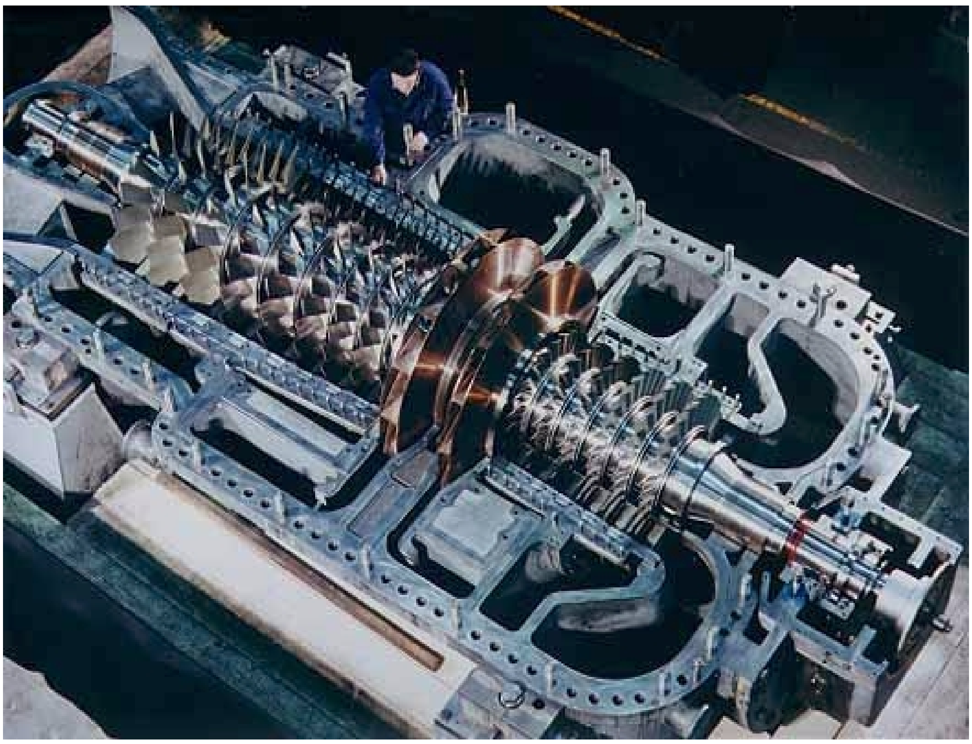
2.1 Introduction

The principal advantage of an axial flow compressor is its ability to delivery very high rates of flow efficiently, albeit at relatively low pressure ratios, compared to other types of compressors such as reciprocating, screw and radial (Bloch 1995, Gresh 1991). Thus axial flow compressors are universally used in aero-engines to provide the necessary compression of intake air prior to combustion and also in certain large-scale industrial applications. For the purposes of this research the industrial application of multi-stage axial compressors is considered.

Typical industrial applications include blast furnaces, refineries, LNG (liquid natural gas) plants, nitric acid plants, aero-engine research facilities, compressed air storage and gas line pumping. Axial flow compressors for these types of application involve inlet flow rates typically in excess of 100,000 m^3/hr and delivery pressure of up to 7 bara (bar absolute). They are physically large machines as indicated in the picture below of a machine under construction.

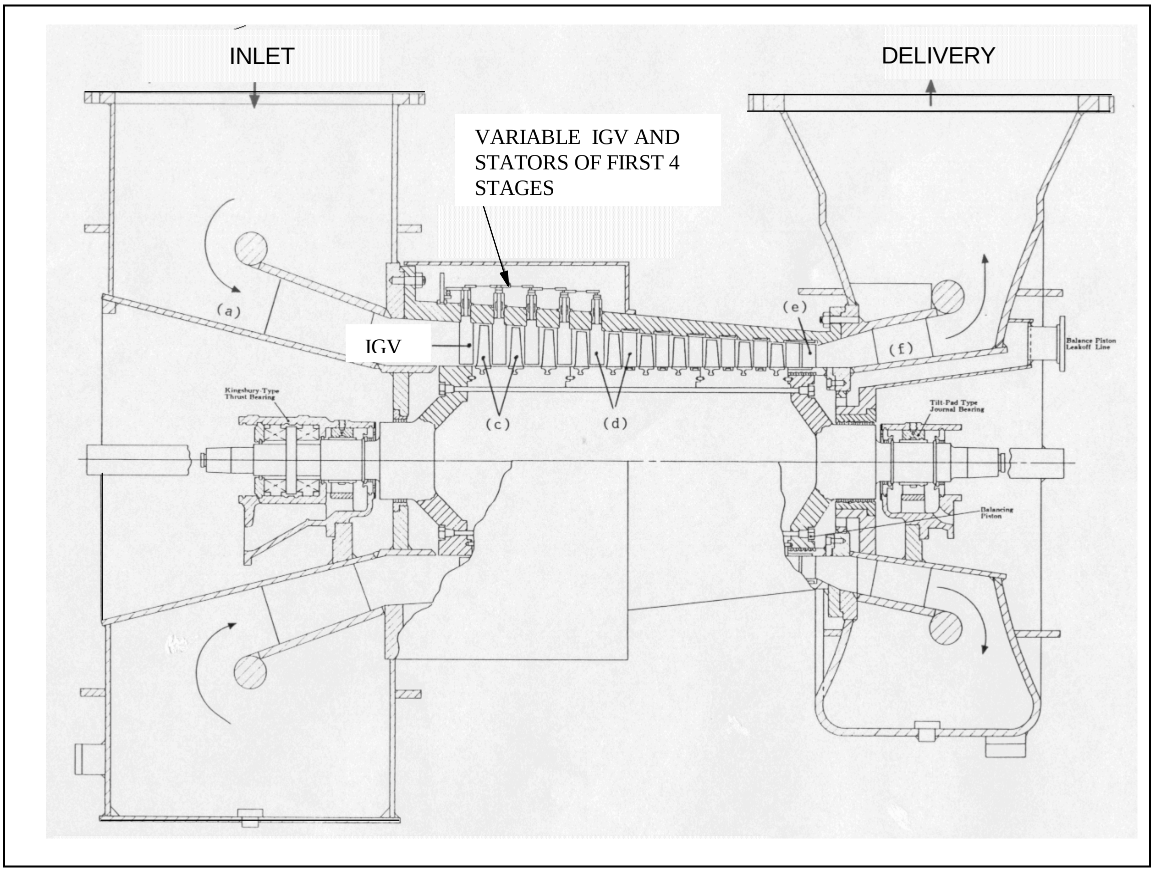
Fig 2.1 Axial flow Compressor for Industrial Application

(Source: Dresser-Rand Corporation, USA)



An axial compressor comprises a number of stages as determined by the required overall pressure rise. Each stage includes a rotating row of blades (rotor) followed by a stationary row of blades or vanes (stator). The rotor blades impart momentum to the fluid thus increasing the total energy of the flow and propelling the fluid along the axis of the machine. The stator vanes convert much of the fluid momentum into pressure energy so that a rise in the static pressure across the stage occurs whilst the mean axial velocity of the flow through the stage is approximately constant. The angles of the blades and vanes relative to the direction of flow are critical to the pressure rise and operating efficiency of a stage.

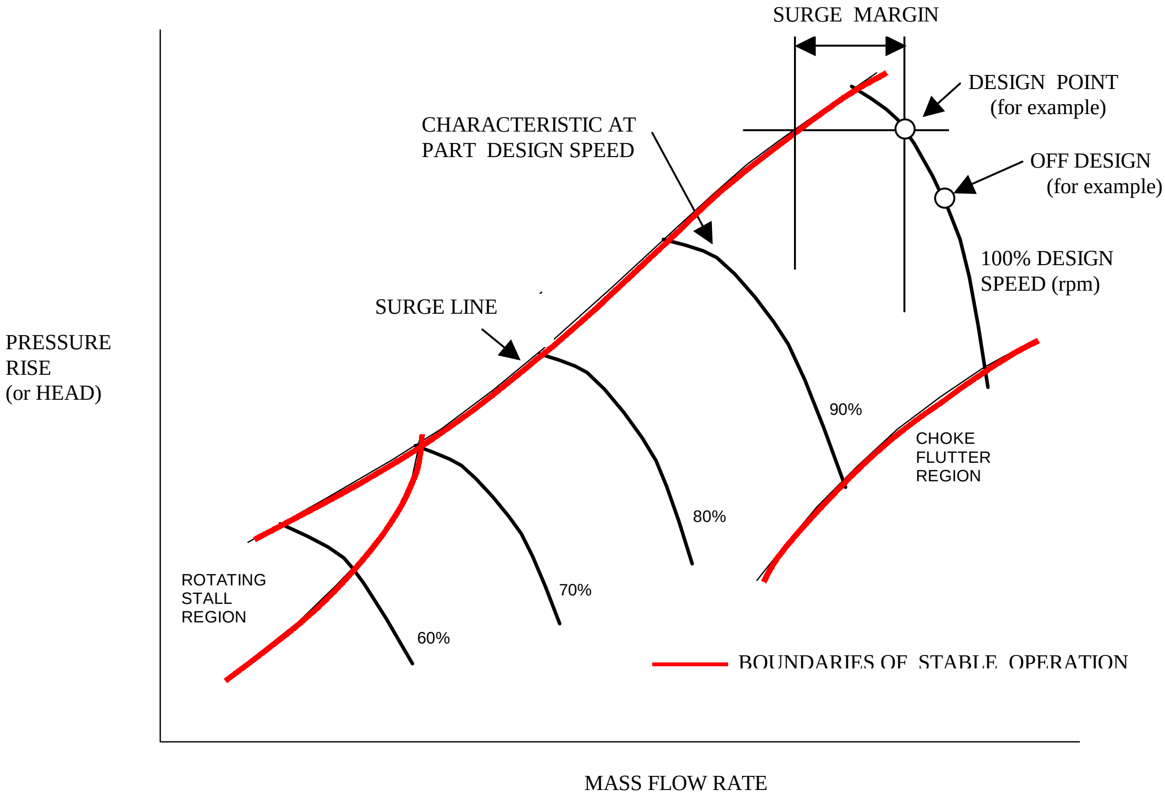
Fig 2.2 Section through Multi-Stage Axial Flow Compressor (Gresh 1991)



Immediately upstream of the first stage in the compressor is a stationary row known as the ‘inlet guide vane’ (IGV). The alternating rotor-stator arrangement is clearly evident in the sectional drawing of an axial compressor shown in Fig 2.2.

Fig 2.3 Typical Compressor Characteristics (based on Gresh 1991)

(Non-Dimensional Axes)



The steady-state characteristics of an axial compressor are usually expressed non-dimensionally in terms of pressure rise and efficiency versus mass flow rate. Typical pressure rise characteristics at different rotational speeds are illustrated in Fig 2.3.

The nominal performance requirements of a compressor in a given application are dictated by the downstream fluid system in terms of delivery pressure and mass flow rate. The downstream fluid system, or simply ‘load’, comprises the network of piping, valves, vessels and equipment through which the process gas flows in achieving the objective of the overall process. Depending on the particular application, it may be required to maintain a particular pressure difference across the load, or alternatively to sustain a particular mass flow rate.

Thus an axial flow compressor is designed to meet the application load requirements consistent with the specified process gas properties, inlet flow conditions and rotational speed of the particular prime mover selected for the application. Operation at this intended operating point is referred to as ‘at design’ whilst operation at other points on the characteristics is referred to as ‘off-design’. At design, axial compressor efficiency is very high, typically approaching 90%. But, as indicated by the steepness of the characteristics in Fig 2.3, the operating range at constant rotational speed is very narrow and is bounded by regions of unstable operation.

If the load resistance is too great (i.e. mass flow rate is too low) then the compressor may enter into either a stall or a surge condition. Stall may occur initially in the form of rotating stall cells (Cumpsty, 1989) i.e. regions of low pressure, low velocity flow, which move circumferentially so that the flow through the machine is no longer axi-symmetric. Surge is a dynamic phenomenon of the total compressor system and refers to a pulsation of the overall axial flow, the frequency of which is determined largely by the volume of the cavities within the machine and immediately downstream. Although stall and surge are strictly different behaviour modes, the boundary at which either might occur is generally known as the surge line (Cumpsty, 1989). Apart from significantly reducing the efficiency of the machine this unstable behaviour may lead to physical damage of the compressor blades. Generally, the onset of surge is difficult to predict and therefore compressor design points are chosen to be well away from the expected surge point, the difference being known as the ‘surge margin’.

At the other extreme of flow, a region of unstable operation is reached in which further reduction in load resistance does not increase the mass flow rate. Gresh (1991) refers to this as the choke flutter region as indicated on Fig 2.3. Operation in this region may result in damaging vibration of the compressor blades.

2.2 Axial Compressor Control

Bloch (1996) describes the application of conventional PID (proportional + integral + derivative) and PI controllers to the main task of compressor performance control. The control element involved may be an inlet or delivery control valve, guide vane positioner, or rotational speed governor. Controlling the operating point of the compressor at a 'safe margin' from the surge line is usually achieved by opening an anti-surge valve to recycle or discharge a portion of the total flow. Again, the controller may use a PI type algorithm for surge control.

Controlling surge by means of a bleed flow (anti-surge) valve as described by Bloch is wasteful of fluid energy. Research has therefore sought either to minimise the losses by developing improved control algorithms when using bleed valves or to avoid losses by using other control elements to modify the flow through the machine. For example, Escuret and Elder (1993) describe the design of a linear optimal controller for active control of surge in multi-stage axial compressors using a bleed valve as the control element. The results of these efforts have generally validated the respective approaches involved whilst identifying difficulties of implementation and generalised application of the particular techniques.

Hosny et al (1991) proposed an active controller using simple proportional closed loop control to effect a dither action of a stator row in order to suppress the small amplitude disturbances which are found to precede the development of large scale compressor instability. A similar approach involving an oscillating guide vane was proposed by Paduano et al (1993). In this experimental application each vane of the IGV is driven independently by a DC servomotor as part of a closed loop control system. The onset of stall is measured in terms of the unsteadiness of the upstream axial velocity and, in response to this input, the vanes are activated to generate circumferential travelling waves of an appropriate phase and amplitude in order to dampen the flow disturbance. The result is that the mass flow rate at which stall eventually occurs is appreciably reduced e.g. by up to 23%. The system arrangement is shown in Fig 2.4 in which it will

Fig 2.4 Active Control of Rotating Stall (Paduano et al 1993)

a) System Arrangement

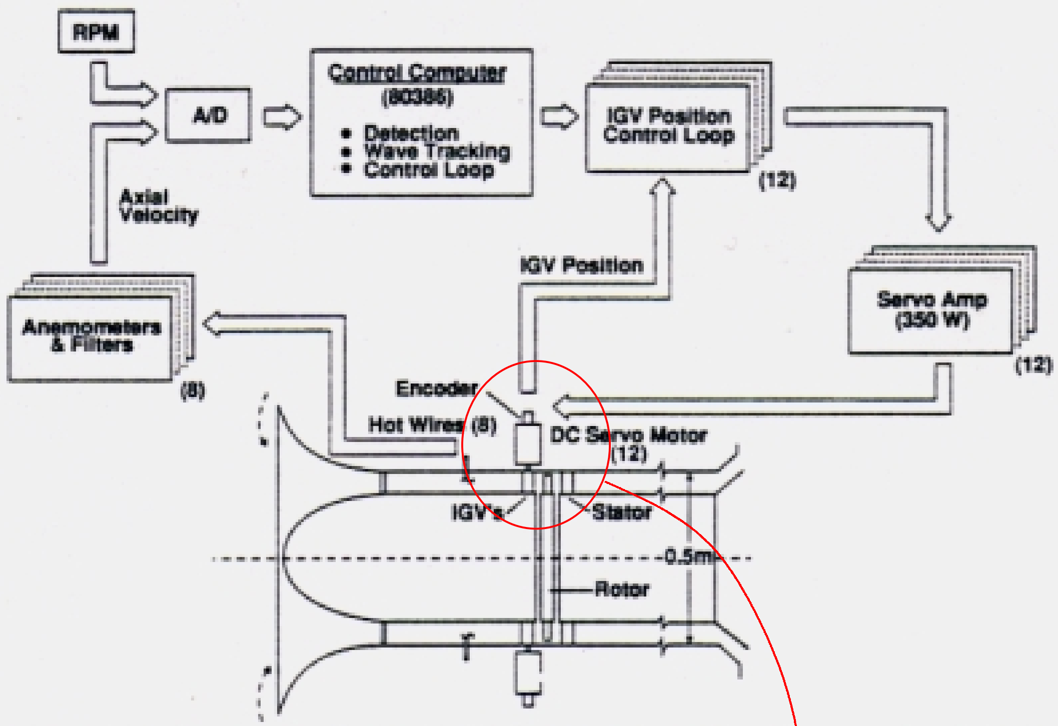


Fig. 6 Hardware components of actively stabilized axial flow compressor

b) Control of Individual IGV vanes

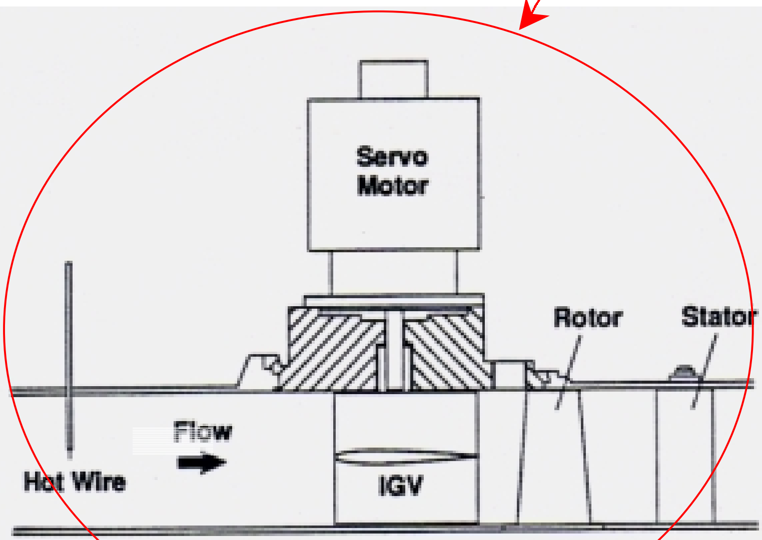


Fig. 4 Compressor flow path

be noted that a central computer controller is employed together with separate vane position control channels.

Extending the stable operating range of axial flow compressors by adjusting stator stagger (angular setting) has been recognised for a long time (McCoy and Hooper 1953) and provision to vary the stator angle on some of the stages of multi-stage machines is common practise. Indeed, it will be noticed that the machine illustrated in Fig 2.2 is fitted with stator adjusters on the IGV and first four stages. Typically, the vane adjustment is made periodically under manual control rather than continually by some form of automatic control system.

Riess and Blöcker (1987) proposed a control system in which all stator rows of an experimental compressor are adjusted automatically to avoid stall conditions. In this system, onset of stall is detected by the occurrence of pressure fluctuations and this initiates rapid adjustment of the stagger angles of the stator rows in order to shift the operating point of the compressor to a more stable location on the pressure-flow map. The experimental machine and one of the stator adjustment devices is shown in Fig 2.5. It will be noted that the vanes in each stator row are driven together by the adjusting mechanism rather than independently. The proposed control strategy relies on the use of pre-determined experimental compressor characteristics and stator settings in order to determine the corrective action necessary for a given actual operating point. It is also proposed that following the initial rapid reaction to move the operating point into a stable region further adjustment of the stators takes place slowly to achieve an operating point at which efficiency is higher. Riess and Blocker's experimental work is based on open-loop control supported by off-line computer calculations but in their conclusion they postulate a fully automatic closed-loop system with microprocessor control.

Although McCoy and Hooper also describe the potential benefits of variable geometry rotor blades in their paper of 1953, the idea does not appear to have been pursued in

Fig 2.5 Fast Guide Vane Adjustment in Axial Compressors
Riess and Blöcker (1987)

a) Experimental Machine with 6 variable stator rows

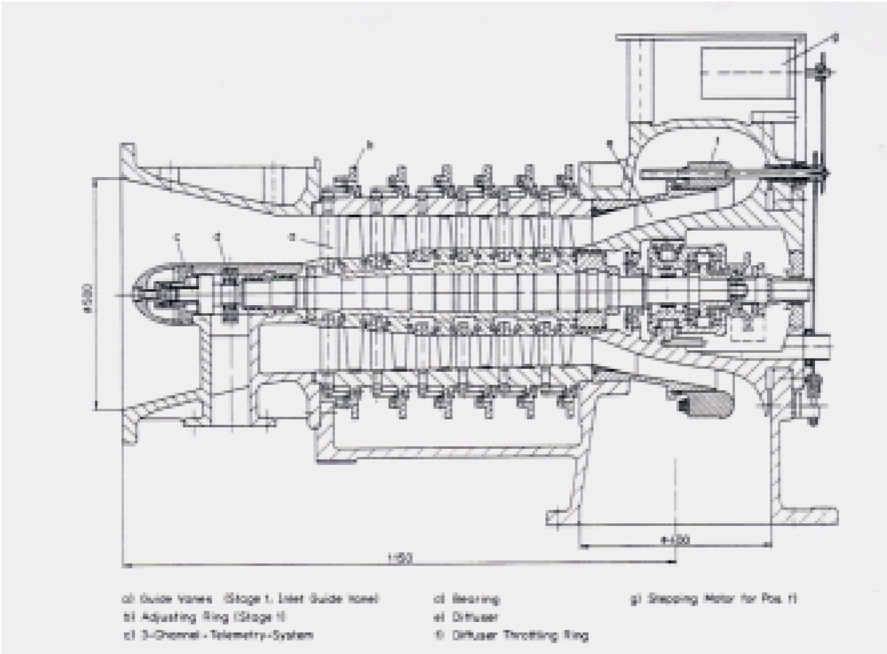


Fig. 2 Six-Stage Experimental Axial-Flow Compressor

b) Adjustment Mechanism for a stator row

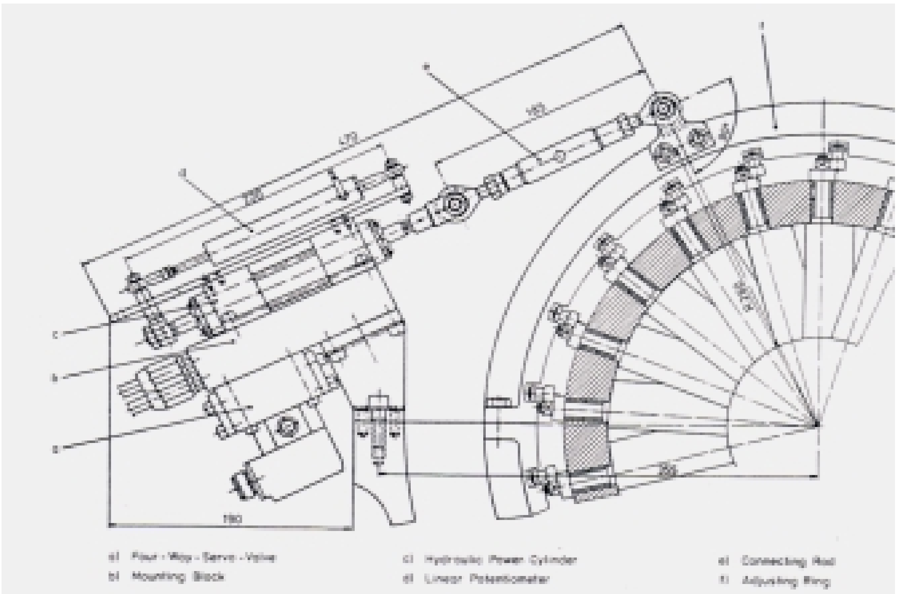


Fig. 15 Fast Acting Guide Vane Adjustment Device (for one Stage)

multi-stage compressor applications probably because of the engineering difficulties involved. Use of variable geometry rotor blades is known, however, in the much simpler and smaller rotodynamic application of automotive VGT (Variable Geometry Turbine) turbo-chargers.

2.3 Intelligent Geometry Compressor

From the foregoing narrative it is clear that there is much interest by researchers and industry in extending and optimising the performance of axial flow compressors and the work of Paduano, and Riess and Blöcker, indicate the potential for achieving this goal through variable geometry stators. In this research the concept of an intelligent geometry compressor (IGC) is postulated in which a significant number, if not all, of the stator rows or individual stator vanes of a multi-stage machine are ‘self-adjusting’ in order to configure the internal aerodynamic geometry to suit the prevailing flow conditions and avoid unstable operation. In this way, the IGC is able to cope automatically with variation in operating conditions and thus provide a considerably larger 'safe' operating range than is achievable by a fixed-geometry machine.

The IGC represents a 'naturally distributed' control application since the physical elements to be manipulated, i.e. stator rows or vanes, are separate entities which are spatially distributed throughout the structure of the machine. Hence a multi-agent system approach is particularly appropriate to the design of an IGC.

CHAPTER 3

Multi-Agent Systems

Over the past two decades a considerable amount of research has been conducted into new and more effective ways of dealing with complex problems that arise in commercial and industrial applications. Underpinning this research is the belief that complexity and uncertainty are better able, sometimes only able, to be addressed by processes which employ human-like cognitive behaviours. Thus a number of 'intelligent' fields of research have emerged during this period of which one of the most promising is Multi-Agent Systems.

In this chapter a general account of multi-agent systems, drawn from published literature in the field, is presented in order to establish the principles involved and serve as a basis for later work. An assessment is also made of how MAS technology relates to other areas of interest namely Intelligent Control Systems and Mechatronics.

3.1 Introduction

The origins of MAS are described by Jennings (1994) in the context of Distributed Artificial Intelligence in which the concept of the agent grew out of early work on blackboards, contract-net and actors. However, the development of similar ideas is reported in other research disciplines. In Control Theory, for example, a subsumption architecture was proposed by Brooks in which overall system behaviour is described as a composite of individual tasks performed reactively by a number of independent processing units (Brooks, 1986). In Computer Science, Shoham (1993) introduced Agent-Oriented Programming as a specialist form of object-oriented programming.

Multi-agent systems (MAS) are decentralised and co-operative problem solving systems. As such they seek to avoid the limitations inherent in centralised control systems when dealing with large or complex applications. Additionally, it is claimed that system problem-solving ability is enhanced due to the combination of multiple problem solving methodologies and sources of information (Jennings 1994). Ferber (1999) refers to this ability as 'collective intelligence' which arises from agent interactions. Multi-agent systems address complexity by decomposing the problem into a number of semi-autonomous entities, called agents (or *intelligent* agents), that communicate and co-operate with one another to achieve the desired goals of the overall system. Agents may be working towards a single global goal or towards separate individual goals that interact in some way (Jennings 1994).

In contemplating a MAS design there are a number of key issues that need to be addressed. These concern the specific architecture of the overall system in terms of the number and type of agents required, their goals and tasks, interconnection and communication methods; the internal architecture and problem-solving capabilities of agents; and last, but not least, how the agents should interact. These issues are inter-related and application-dependant, but much research has been carried out to define generic approaches and models on which specific solutions may be based. Reference to this research is made in the following sections.

3.2 Agents

An agent may be an entity of software, hardware or a combination. Many different types of agent are reported in the literature, differentiated by behaviour and internal structure.

3.2.1 Agent Classification

Ferber (1999) proposes a classification scale based on the capacity of agents to accomplish complex tasks individually. The scale ranges from purely *cognitive* agents at one extreme through to purely *reactive* at the other. These extremities reflect two basic schools of thought amongst researchers in multi-agent systems.

The first school, favoured by Brooks (1990) and Ferber (1999) proposes systems of large populations (e.g. > 100) of simple reactive agents which have limited internal states and which rely on collective behaviour to tackle complex tasks. Other researchers, Wittig (1992) and Jennings (1994) propose systems involving a small number (e.g. < 10) of cognitive agents which are internally much more sophisticated, are able to reason about their environment and are much more autonomous in their actions.

Purely reactive agents are also referred to as *situated* (or situated-action) agents, Jennings and Wooldridge (1998), meaning that they are positioned in their environment and communicate only by the propagation of signals within the environment rather than by message exchange with other agents. Agents that exhibit both reactive and cognitive capabilities are termed *hybrid.*

Other agent classifications that appear in the literature are more application oriented. For example, Shen et al (2001), in addition to cognitive, reactive and hybrid agents, identifies the following types:

a) *software agent* – as distinct from human and hardware intelligent agents
b) *mobile* - software agents capable of moving from one machine to another
c) *interface* – agents which act as intermediaries between a human user and an automatic system
d) *intermediate* – agents which provide specialist services to other agents for example, agents which serve as brokers, facilitators, mediators or matchmakers.

Jennings and Wooldridge (1998) offer a similar classification.

3.2.2 Attributes

The range of attributes possessed by agents varies according to agent type and application. Researchers generally agree however that for an entity to be regarded as an agent it must possess, to some degree, attributes of autonomy and co-operation as a minimum. More generally, Ferber (1999) defines an agent as a physical or virtual entity possessing the following attributes:

a) capable of acting in an environment
b) able to communicate directly with other agents
c) is driven by a set of tendencies (in the form of individual objectives or of a satisfaction/survival function which it tries to optimise)
d) has resources of its own
e) capable of perceiving its environment (but to a limited extent)
f) has only a partial representation of its environment (and perhaps none at all)
g) possesses skills and can offer services
h) may be able to reproduce itself
i) tends towards satisfying its objectives, taking account of the resources and skills available to it and depending on its perception, its representations and the communications it receives.

Khosla and Dillon (1997) offer a similar range of attributes in their ‘PAGE’ (Percept, Action, Goal and Environment) description of an agent. Other researchers offer less comprehensive lists and Jennings and Wooldridge (1998) define a minimal list of just three attributes: *autonomy*, *learning* and *co-operation*.

3.2.3 Modules

Naturally, the constituent modules of agents correlate with the range of attributes that they possess. As with attributes, there is a minimum sub-set of modules and these are identified by Rzevski (1994) and Shen et al (2001) as *perception*, *cognition* (or reasoning) and *execution* (action). Shen et al proceeds to identify additional modules that may be included within the internal structure of an agent:

a) communication interface
b) social knowledge
c) self knowledge (self representation)
d) domain knowledge (domain representation)
e) knowledge management
f) learning

g) problem solving methods
h) co-ordination
i) planning and scheduling
j) control
k) conflict management
l) application interfaces

It may be argued that the model of perception, cognition and execution is a valid generic representation for any agent since the items in the above list could quite easily be grouped under these three headings.

3.2.4 Architecture

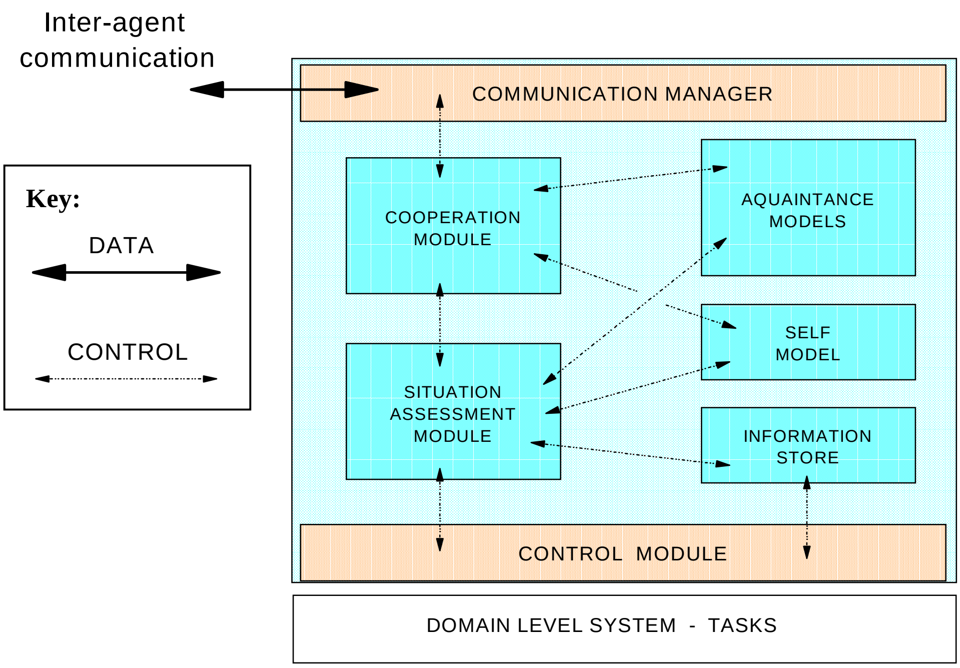
The architecture of an agent defines the internal organisation and interconnection of the constituent modules necessary to achieve the required agent behaviour. Agent architectures are thus linked to agent type and may be classified by behaviour or alternatively by the type of organisation structure. Given the potentially wide variety of agent behaviour many types of agent architecture are possible and, indeed, many different descriptions appear in the literature. Three widely recognised architectures are briefly described below by way of example.

a) Hybrid Agent Architecture

The GRATE agent architecture was devised by Jennings (1992) as a generic model for use in the development of multi-agent applications. Shen et al (2001) describes it as a collaborative architecture, which also incorporates deliberative (cognitive) features and thus classifies GRATE as a *hybrid* architecture. GRATE basically comprises four main modules and a database of reference models as illustrated in Fig 3.1.

`

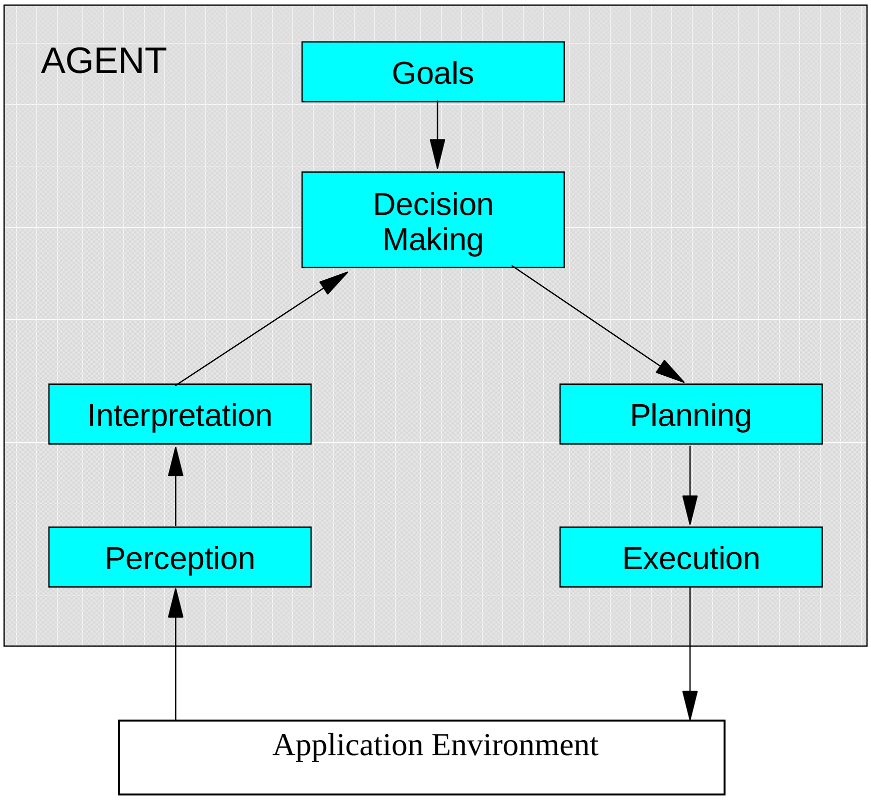
Fig 3.1 GRATE – An example of a hybrid architecture



b) Modular Agent Architecture

This type of architecture is widely used in multi-agent systems and may range from very simple comprising a few modules to complex organisations involving a large number of modules. It is sometimes referred to as a horizontal-module architecture since the modules are at the 'same level' in the organisation. Also, in this type of architecture, all of the connections between the modules are typically fixed i.e. the information flow is pre-defined by the agent designer. The simple example shown below reveals the basic perception, cognition and execution structure.

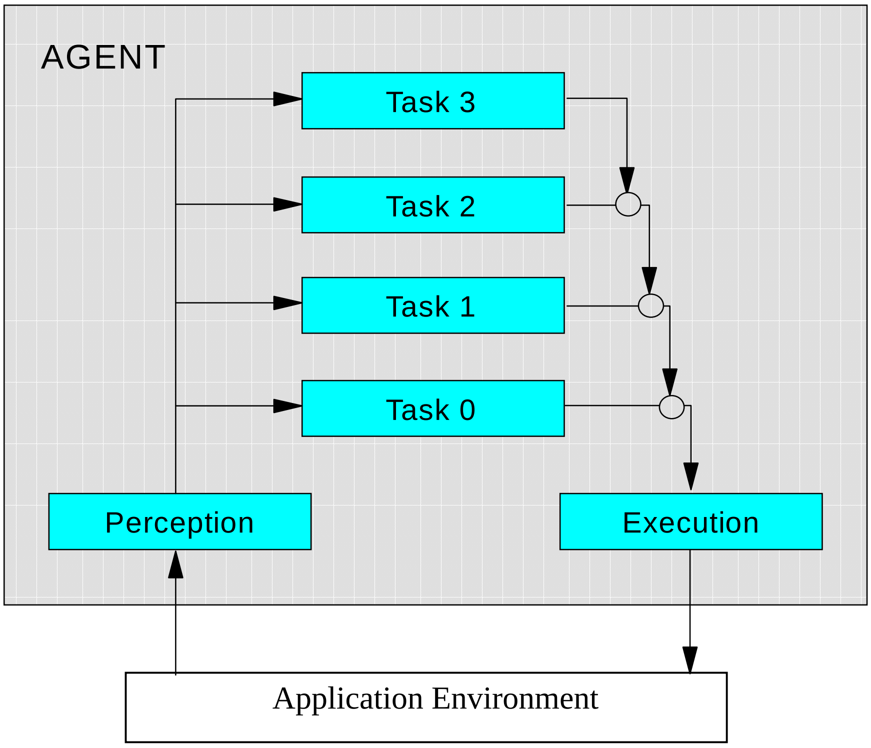
Fig 3.2 Example of a modular agent architecture



c) Subsumption Architecture

This is a special architecture first proposed by Brooks (1986) for reactive agents. It is basically a modular architecture but instead of horizontal linking between modules, here the modules are organised vertically. The modules operate in parallel, with those higher up in the organisation having a dominance over those lower down. This means that the higher modules can inhibit the behaviour of lower level modules. As with the modular architecture, the designer defines the connections between modules and also, in this case, the dominance relationships that exist between them. The subsumption architecture has been successfully used in simple robotic applications e.g. AGVs (automated guided vehicles) and although intended for reactive agents it is evident that it would also be possible to use it for cognitive agents. A typical example is shown in Fig 3.3.

Fig 3.3 Example of a subsumption agent architecture



3.3 Systems of Agents

3.3.1 Architecture

Agent system architectures provide the organising frameworks within which agents are designed and constructed. Three general types of architecture are referred to in the literature and are briefly considered here.

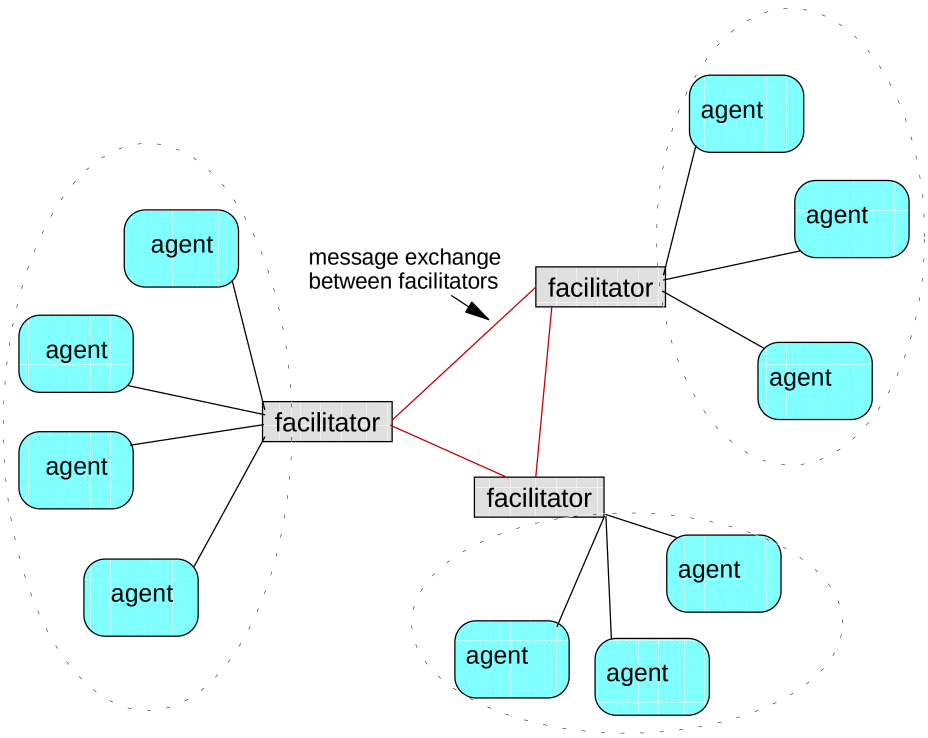
a) Hierarchical Architecture

Generally, hierarchical architectures are not favoured because of their centralised character and the well-known disadvantages associated with centralised systems. However, where the application environment is organised hierarchically then it has been useful to organise the multi-agent system in the same way. The ADEPT (Advanced Decision Environment for Process Tasks) (Norman et al 1997) architecture for multi-agent systems provides an example of a hierarchical architecture developed for agent-based industrial systems. It is a nested structure of agencies each comprising a responsible agent plus a set of subsidiary agencies.

b) Federated Architecture

This type of architecture is increasingly being considered as an alternative to hierarchical architectures for large industrial agent-based applications. In a fully federated agent-based system there is no explicit shared facility for storing active data; rather the system stores all data in local databases and handles updates and changes through message passing. There are several ways in which a federated architecture may be configured. The example shown in Fig 3.4 uses 'facilitator' agents to manage communication and co-ordination between sets of agents (McGuire et al, 1993). In other schemes this role is carried out by 'broker' agents or 'matchmaker' agents as described by Shen et al (2001).

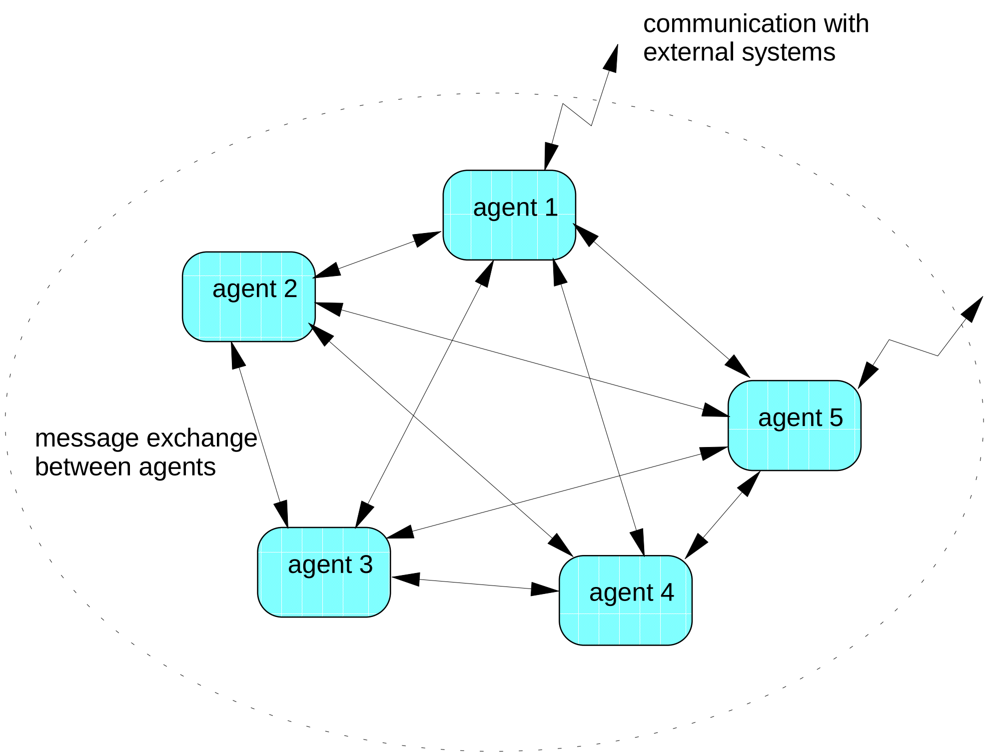
Fig 3.4 Federated Architecture for MAS



c) Autonomous Agent System Architecture

This approach, also known as Agent Network, relies upon the autonomy of the agents involved since neither communication nor state knowledge is consolidated within the architecture. Numerous researchers have used this approach to develop agent-based concurrent design and manufacturing systems and other industrial-based applications. AARIA (Parunak et al 1998) is a multi-agent system for factory application in which manufacturing entities are encapsulated into autonomous agents. The ARCHON research (Wittig 1992) is another example in which a relatively small number of cognitive agents manage an electricity distribution network. Shen et al (2001) suggests that the agent network is especially useful for autonomous robotics control. The simplicity of the architecture is shown in Fig 3.5.

Fig 3.5 Autonomous Agent System Architecture (Agent Network)



3.3.2 Communication

The two main types of communication used within multi-agent systems are 'shared memory' and 'message passing'. The most widespread example of the former is the blackboard system (Englemore and Morgan 1988). The blackboard itself is a global database containing entries generated by the agents. The entries include intermediate results generated during problem solving and include both elements of the problem solution and information deemed important in generating solution elements. Blackboard systems have been widely used in research with the HEARSAY II (Erman et al 1980) speech understanding research being an often-quoted example.

Message passing ideas have been drawn from conventional object-oriented programming and in particular from object-based concurrent programming. This is the approach adopted for a number of industrial applications that have been reported in the literature (Khosla and Dillon 1997, Wittig 1992, Jennings 1993). Message passing has some advantages over the blackboard system. In particular, shared memory systems generally do not scale up well - a single blackboard can be a severe bottleneck and multiple blackboards have the same semantics as message passing systems.

In addition to the above, two other levels of communication are recognised. The first is a primitive form applicable to communities of simple reactive agents in which communication is effected by propagation of signals within the environment (Ferber 1999). In this case the communication process may be regarded as *incidental* rather than *intentional.* At the other extreme of sophistication is the use of formal languages involving extended exchange of series of messages to support a 'conversation' between agents (Weiss 1999). In this context there is much, and growing, interest in the field of *ontology* as a possible mechanism for agents to share the meaning of exchanged symbols (Shen et al 2001).

Modes of communication are *direct* as in the case of message exchange or *indirect* as in posting to a blackboard. Agents may communicate to a selected agent, *point-to-point*, to a selected group of agents, *multi-cast*, or to all agents, *broadcast*. In addition, communication may be *synchronous* or *asynchronous.*

There are a variety of protocols and languages for supporting both the communication linkages and the exchange of information between agents and a number of organisations are developing standards for this purpose e.g. FIPA (The Foundation for Intelligent Physical Agents), Weiss (1999). At the lowest level of inter-agent communication local area network protocols for high speed serial links apply such as DeviceNet and CAN, an adaptation of the Intel-Bosch car area network protocol widely used in automotive applications. At the next level up, CORBA (Common Object Request Broker Architecture) is a standard which defines a mechanism by which objects written in different languages and executing in a distributed environment can make requests of, and respond to, one another. Standards which support a 'conversational' level of communication between agents include KIF (Knowledge Interchange Format) and KQML (Knowledge Query and Manipulation Language) Finin et al (1993).

The rapidly developing field of local wireless communication clearly holds great potential for multi-agent system applications. Office-based applications of wireless LANs (local area network) already exist based on Bluetooth technology and industrial applications are expected to grow as hardware costs inevitably fall, Allan (2001).

3.3.3 Interaction between agents

Three forms of interaction between agents exist namely co-operation, co-ordination and collaboration. These forms are closely related but represent quite distinct concepts.

Co-ordination may be regarded as the process by which agents ensure that their individual actions are consistent with the overall goals of the system. Basic mechanisms for co-ordination include: -

a) mutual adjustment – agents share information and resources to achieve some common goal by adjusting their behaviour according to the behaviour of the other agents
b) direct supervision – one agent has some degree of control over others which may have been arrived at through mutual adjustment
c) standardisation – supervisor agent establishes standard procedures for agents to follow in given situations
d) mediation – one agent serves as a facilitator or broker to influence interaction between agents

Co-ordination techniques, which may employ these mechanisms, are organisational structuring, subcontracting, negotiation and multi-agent planning.

Collaboration arises when one agent is able to perform a task, which only it can do, and as a result enables another agent to achieve its own goal. Clearly the need for collaboration is determined by the allocation of skills and resources to agents made when the system was designed. Except in simple cases, it is often necessary to co-ordinate collaboration in order to make effective use of skills and resources consistent with overall system goals.

Co-operation is about agents' actions being mutually supportive to their respective goals. Supportive action by one agent for another may be intentional or incidental. Ferber (1999) states that, put simply, the problem of co-operation condenses down to determining who does what, when, by what means, in what way and with whom. Ferber goes on to summarise this in the formula:

Co-operation = collaboration + co-ordination of actions + resolution of conflicts

Ultimately, the co-operation strategy of a multi-agent system is critical in ensuring that actions by autonomous agents in pursuit of local goals have, at least, a beneficial, if not optimal, effect on overall system performance.

3.4 MAS Applications

A wide range of instances of multi-agent system application is reported in the literature. Inevitably, given the relatively short history of the field, the applications cited are mostly experimental or research-based projects.

Of those relating to industry, most of the recent applications are concerned with manufacturing enterprise control and planning. For example, the ISCM (Integrated Supply Chain Management) system manages the flow of material through the organisation by means of a network of co-operating intelligent agents (Barbuceanu and Fox 1997). The system comprises several types of cognitive agents which communicate using the high level language KQML. Another example in this area is MetaMorph II (Shen et al 1998), which uses a hybrid agent-based architecture to integrate design, planning, scheduling and other activities in the manufacturing enterprise. Notably, in these industrial systems the entities involved are software agents. An exception is found in Holonic Manufacturing System applications where agent (or holon) representation is extended to physical entities such as machines, products and mobile robots.

Few applications relating to control of industrial hardware are reported beyond some early examples. These include the ARCHON research (Wittig 1992) which implemented multi-agent systems for control of a particle accelerator and for the monitoring and fault detection of an electricity distribution network. Other research projects have applied MAS to co-ordinate the actions of individual robots (Mataric 1994, Steels 1994) and the approach has been extended and applied to collision avoidance of vehicles. In particular, Jennings and Wooldridge (1998) describe a system for air traffic control called OASIS, developed in 1996, in which agents are used to represent both aircraft and traffic controllers.

Of particular interest to this research is the application area described by Ferber (1999) as ‘cellular robotics’ which relates to building robots on a modular basis. For example, a manipulator arm has been modelled as a multi-agent system with each element of the arm considered as an agent (Overgaard et al 1994). Currently, most of the interest in multi-agent systems appears to focus on internet applications for information filtering and gathering and e-commerce.

3.5 Design of Intelligent Machines

In the context of intelligent machines there are two established areas of technology, which appear in the literature. These are Intelligent Control Systems and Mechatronics. It is appropriate therefore to review these areas briefly and to consider how they relate to multi-agent systems regarding the design of intelligent machines.

3.5.1 Intelligent Control Systems

Reference to Gupta and Sinha (1996), Lu (1996) and Antsaklis and Passino (1993) reveals that Intelligent Control is the latest phase of development in the field of Control Engineering. This field is generally concerned with the problems of controlling physical equipment, plant and machines of the type of interest in this research. Control Engineering has evolved in distinct phases over time as indicated in the chart of Fig 3.6, which draws on material from the three sources, referenced above. This is not to suggest that the ideas and methods of one era have totally replaced those of the previous era. Rather, new developments have been absorbed into the general body of knowledge to be deployed alone or in combination with established methods depending on application.

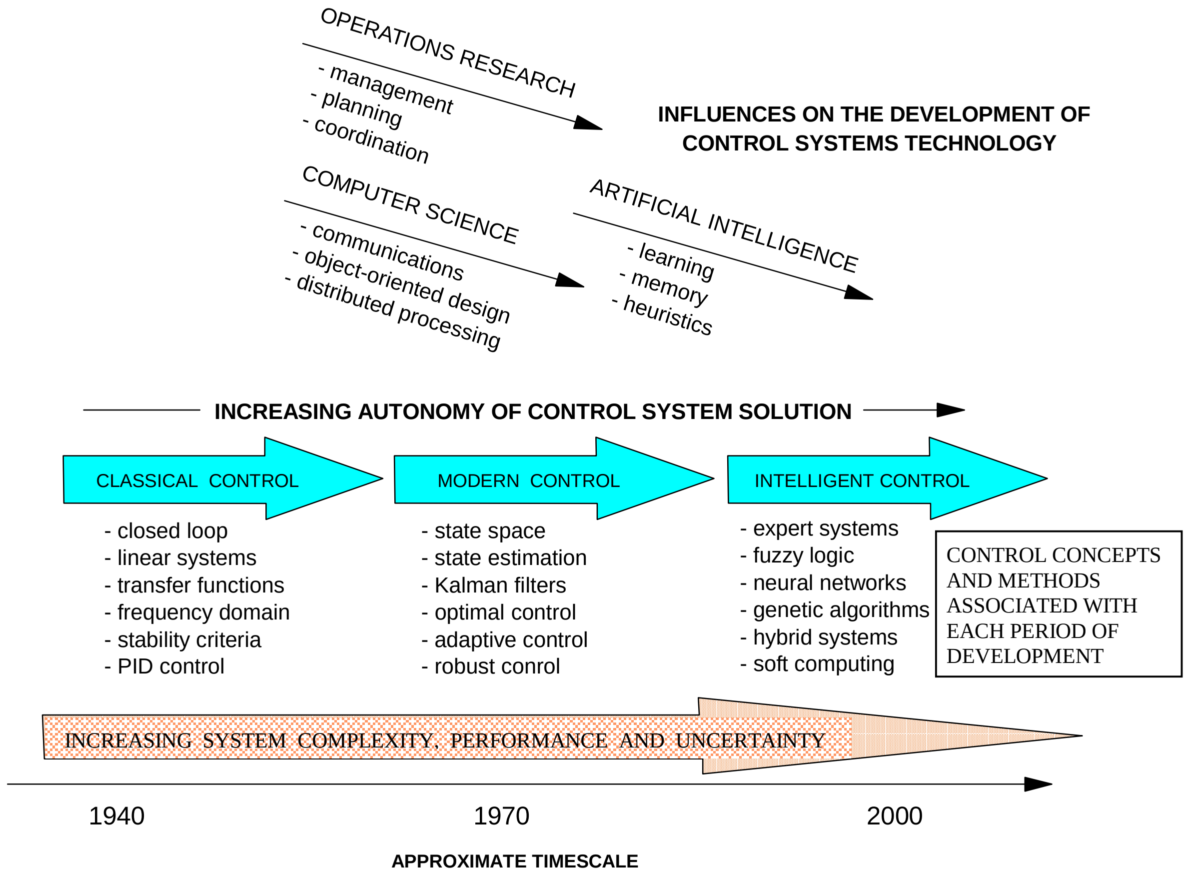
The emergence of Intelligent Control has been fuelled by three major needs (Antsaklis and Passino 1993):

i) The need to deal with increasingly complex dynamical systems

ii) The need to accomplish increasingly demanding design requirements

iii) The need to attain these design requirements with less 'a priori' knowledge of the plant and its environment, that is, the need to control under conditions of uncertainty.

Fig 3.6 Overview of Control Engineering Evolution



Interest in developments in Artificial Intelligence grew in the general belief amongst researchers in Control Engineering that methods based on human cognition are better able to deal with these needs than those of conventional, i.e. classical and modern, control technology. A controller was sought which was able to reason under conditions of uncertainty and take actions without human intervention i.e. an autonomous controller. Therefore, the definitions of Intelligent Control found in the literature, whilst expressed differently, generally agree that the field encompasses those systems which exhibit human-like behaviours in one or more of the control techniques which they employ. Examples of human-like behaviour are parallel information processing, adaptation to environment, associative memory, learning, generalisation and self-organisation (Lu, 1996).

As the chart of Fig 3.6 indicates, Intelligent Control embraces a number of concepts and methods from Artificial Intelligence. Through the 1990s, the four main intelligent methodologies applied to real problems were: expert (knowledge-based) systems, fuzzy logic control, artificial neural networks (ANN) and genetic algorithms. Initially, these methods were applied separately being used where a mathematical model of the application process was either unavailable or so complex as to be impractical to implement. Later, intelligent methodologies were developed based on a combination of the earlier tools. Examples of these are ‘neuro-fuzzy control’ (Jang and Sun, 1995) and, more recently, ‘soft computing’, which combines genetic algorithms, fuzzy logic and neural networks (Muscato, 1998).

A common approach to the organisation of intelligent control systems in complex applications, such as robotic systems, is described by Antsaklis, Lemon and Stiver (Gupta and Sinha 1996). This uses a hierarchical architecture with three main functional levels viz., organisational (top level), management (or co-ordination), and execution (lowest level) and is based on the principle, put forward by Saridis (1979), of increasing intelligence towards the upper level of the hierarchy. Such systems are often described as *hybrid* in that they comprise a number of separate control units, some of which incorporate an intelligent control method whilst others employ methods based on conventional, i.e. classical or modern, control concepts. The control units at the execution level operate to some degree autonomously with regard to local goals but are constrained by the co-ordinating action of the more intelligent units in the upper levels of the hierarchy in order that overall system goals are achieved. In effect, this approach resembles a form of centralised control.

There are two general observations about Intelligent Control systems that may be made from this brief introduction. Firstly, intelligent behaviour of the overall system relies on the deployment of intelligent methods within one of more of the constituent control units rather than on any collective effect of the autonomous behaviour of all control units in the system. Secondly, Intelligent Control system design focuses exclusively on the logical and functional behaviour of the control system and does not directly aim to influence the physical design of the plant or machine being controlled.

3.5.2 Mechatronics

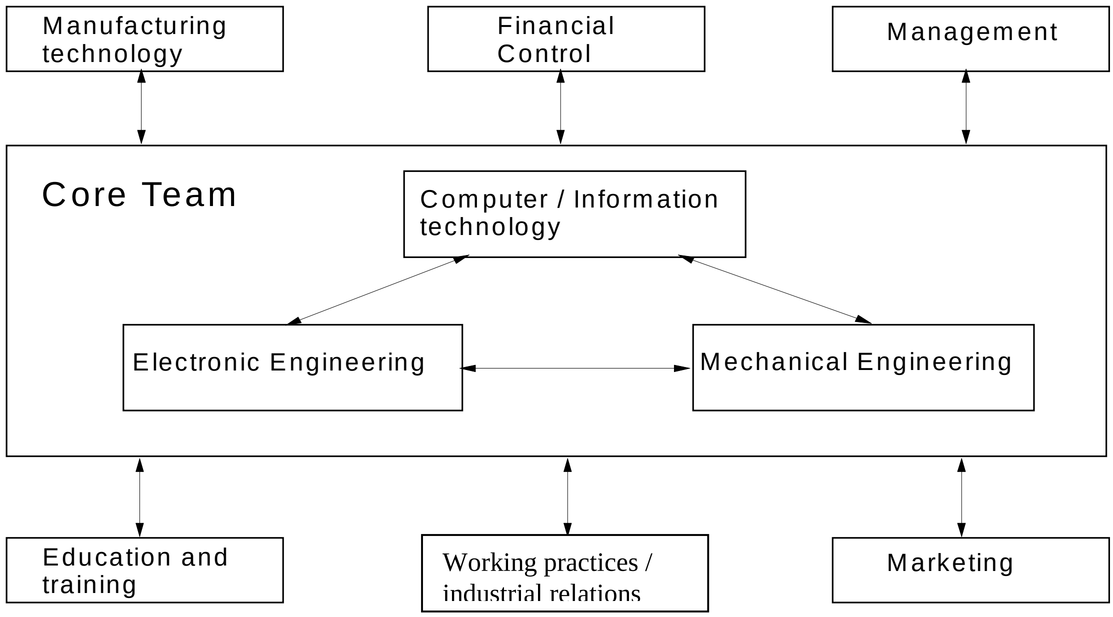
Mechatronics emerged as an engineering discipline in the 1970s in response to the need for a more effective way in which to design mechanical products containing embedded microcomputer technology (Rzevski, 1994).

The continued rapid development in electronic hardware, communications and software methods has resulted in computer and information technology becoming the dominant technology in driving the enhancement of existing products and the innovation of entirely new products. Thus the objective of Mechatronics is to provide a framework for engineering design, from concept to manufacture, within which this dominant technology can be effectively integrated with electronic and mechanical disciplines to achieve potentially complex products, yet capable of low cost production. Contemporary examples of 'mechatronic products' include consumer goods such as DVD players, cam-corders and industrial products such as robotic modules and 'smart' sensors.

The organisation and management of the design team is a key part of the mechatronics approach. This must involve specialists from all of the disciplines that have a role to play in the product life cycle but with the main design work being performed by a core team of engineers as illustrated in Fig 3.7.

In order to achieve the objective of Mechatronics, Bradley et al (1991) suggest a top down and information based strategy in which the overall system is broken down into a series of blocks or modules in order to facilitate further analysis and design. Typically this will include: environment, measurement, communications, processor, software, actuation and interface modules. Depending on the complexity of the application, each of these modules may be broken down to another level. Within each module, subsequent conceptual and functional design may identify elements that are sufficiently self-contained, functionally and physically, as to be treated as mechatronic systems in their own right. In which case, the process of decomposition is applied to these systems in the same way. Eventually, detailed design of components in the system can be undertaken.

Fig 3.7 The Elements of Mechatronics (Bradley et al 1991)



The design approach described above, when applied to a complex system application, would be expected to result in a number of 'mechatronic' units, each containing integrated processing and communications capability. Thus, for such applications, Mechatronic solutions tend to be distributed. However, Mechatronics, of itself, does not predicate that the end product or system is intelligent. The behaviour of the end product will be determined by the control strategies and methodologies selected by the design team in the conceptual phase of the design process.

3.5.3 Relationship between MAS, Intelligent Control and Mechatronics

MAS and Intelligent Control share the objective of achieving autonomous systems capable of sustaining performance under conditions of uncertainty. However, as explained in the preceding sections of this chapter, the respective approaches to achieving this goal differ. A multi-agent system achieves intelligent behaviour of the overall system through the cooperative action of all constituent agents, whereas in Intelligent Control, the system autonomy derives from the intelligent behaviour of selected control units operating, typically, within a hierarchical architecture. Either approach may be adopted for the conceptual design of an intelligent machine, but one may offer more advantage than the other in certain applications.

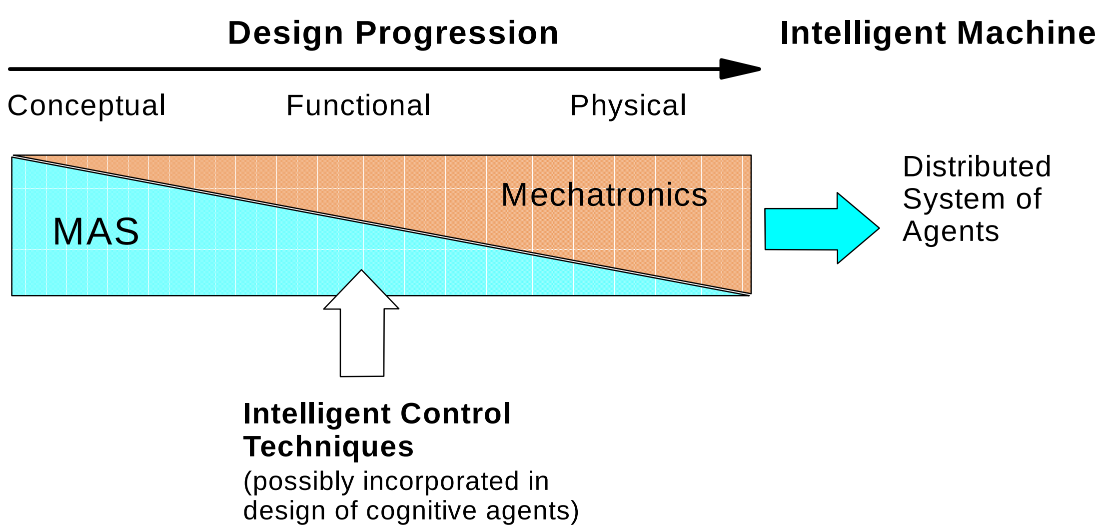
For example, where the application is naturally distributed or if the number of discrete control entities is large then an MAS approach provides the more effective solution since the hierarchical organisation of an Intelligent Control system will limit the rate of information flow (bandwidth) between control units and adversely affect system performance. For a similar reason, the autonomous behaviour of agents also offers advantage in those applications requiring a purely reactive system solution. For applications involving relatively few control entities and in which it is critical to achieve optimal overall performance, the formal control structure and methods of Intelligent Control may provide a more reliable solution. The equivalent MAS would need to employ a sophisticated co-operation strategy to achieve the required performance.

Considering the constituent elements of the respective systems, there is potential similarity between 'control units' and 'agents' in both the internal architecture and the methods employed. For example, cognitive agents of the type used in the Archon application (Wittig 1992) are of a similar conceptual design to the control units used at the co-ordination level of an Intelligent Control hierarchy (Antsaklis and Passino 1993). In general, however, it would be expected that an 'agent' would be of simpler design, but more numerous, than a 'control unit' of an equivalent system.

Either MAS or Intelligent Control could be used in conjunction with a Mechatronics approach to design an intelligent machine. However, there appears to be much more synergy between MAS and Mechatronics. Firstly, a multi-agent system is always distributed and this, as observed above, is the natural form of a mechatronic system solution. Secondly, the emphasis on physical decomposition in MAS makes it possible for agents to be conceived as integral parts of an intelligent machine and therefore be realised directly as mechatronic units. In contrast, the functional hierarchy of an Intelligent Control system is less likely to map directly onto a mechatronic solution and thus would restrict the options for physical design.

The strong synergy between MAS and Mechatronics suggests an overall strategy for the design of intelligent machines in which MAS provides the conceptual basis for intelligent behaviour and Mechatronics the means of faithfully realising the concept in the end product.

Fig 3.8 Design of Intelligent Machines



This strategy is shown in Fig 3.8 which shows the greater influence of MAS during the conceptual phase of design whilst the role of Mechatronics becomes more dominant as the machine design progresses. For completeness, the potential inclusion in the MAS design of Intelligent Control methods is also shown.

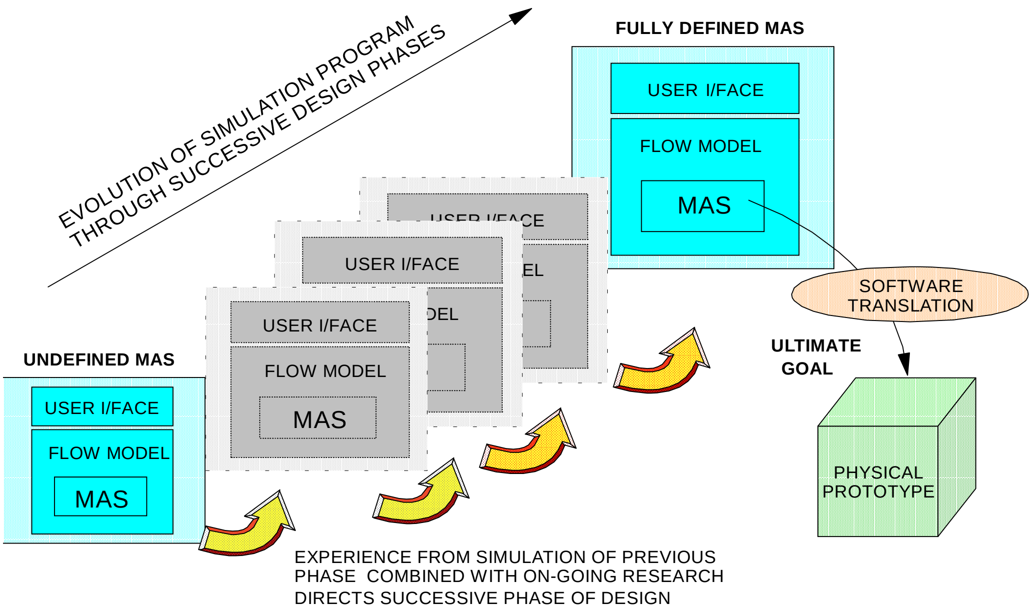
CHAPTER 4

Research Methodology

As previously explained, the intelligent geometry compressor (IGC) is conceived as a multi-agent system and the main goal of the research is to design this system sufficient to demonstrate the capability for enhanced machine performance.

In general terms, the methodology adopted for achieving the research goal was based on the notion of a computer simulation of the IGC application which evolves through a number of discrete phases of development as shown in Fig 4.1.

Fig 4.1 Research Methodology



For each phase, the simulation reflects the knowledge of the total system at that point in the design process. In the first phase of IGC design the compressor flow model was combined with an MAS represented by a single agent of limited functionality. By analysing the results and observations of simulation trials, the behavioural and functional requirements for the agent system were revealed and the design of the multi-agent system was able to proceed. In the next phase, the initial design of MAS was introduced into the simulation along with appropriate changes to the flow model and user interface and the process of trial and evaluation repeated. The process continued through subsequent phases until overall performance goals were achieved and the agent system was fully defined and modelled.

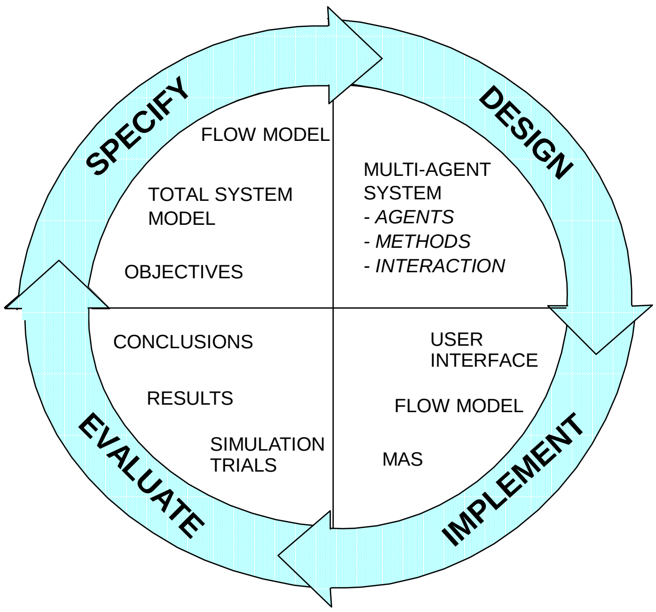
The methodology described is similar in concept to an approach proposed by Guida and Tasso (1994) for the design of knowledge-based systems in which a prototype system, initially ‘empty’, is developed incrementally in a cyclic process.

The main advantages of the methodology are:

1) The insight and knowledge gained through simulation at each phase provided direction for subsequent phases.
2) The process proceeded in parallel with, and benefited from, other research activities.
3) A working simulation program was available from an early point in the research which enabled progress to be demonstrated and reviewed.

Within each phase of the overall approach, a lower level methodology was applied involving a cycle of steps, which included the design of the agent system and the development of the computer simulation program. Within each of these steps specialist methodologies and tools were applied appropriate to the specific tasks e.g. object-oriented design for software. This cyclic process is shown in Fig 4.2

Fig 4.2 Design Cycle



Another result of this methodology is that the agent system description is eventually available in the form of computer source code. Indeed, the code for the software part of the agent system may be regarded as a prototype of an eventual real-world system. This depends only on the degree of compatibility between the computer system used for simulation and that which might be selected for a target system.

The number of phases required to complete the design could not be predicted at the outset. In the event, this research was carried out in three main phases described in later chapters. Before commencing the first phase of the design it was necessary to consider some general points relating to the simulation program design. These are dealt with in the next chapter.

CHAPTER 5

Simulation Program Design

Preliminary Considerations

There are a number of proprietary software packages available such as MathCad and MatrixX which support the development and simulation of engineering control systems. However, in order to maximise flexibility for modelling of the agent system it was decided to develop a custom program using an appropriate high level language and therefore the standard package alternatives were not seriously considered.

Given the intention to develop a custom program, there were a number of general design issues concerning program organisation and representation of agents that were decided before commencing the main task of the research. Firstly, the details of the particular computer system used for simulation program development are given.

5.1 Computer System

The computer is a Pentium PC running Windows 95 operating system. Application software included Microsoft packages for graphics, spreadsheet and text. For program development, Borland C/C++ Integrated Development Environment was installed. This enables C and C++ code to be created with full access to Windows functions through an Application Program Interface (API).

The particular features of the computer system of interest for the planned simulation work are summarised below:

a) C++ supports object classes.

b) Windows 95, being a multi-tasking operating system, supports both processes and threads as means of achieving 'apparent' concurrent operation. (True concurrency is achievable only with a multi-processor system).

c) A variety of mechanisms are available for synchronising and passing data between concurrent processes and threads.

d) The C/C++ source code is compilable to produce fast executable code.

e) A wide range of familiar Windows user interface controls can easily be incorporated.

f) The user interface can be enhanced by the inclusion in the executable code of custom bitmap graphics conveniently created in the available graphics packages.

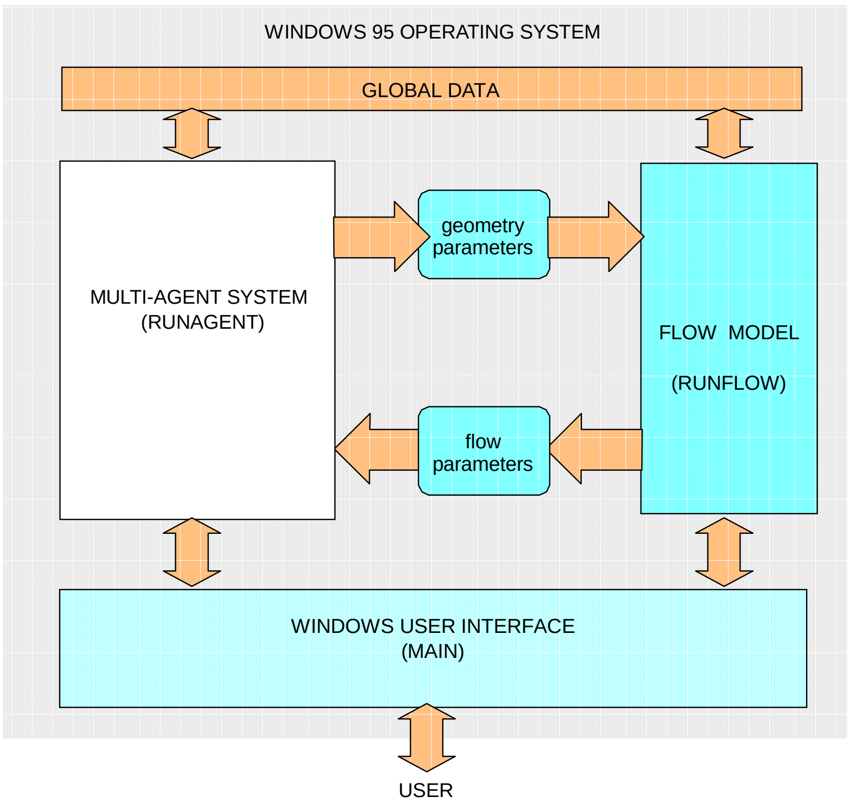
g) Data can be captured during simulation runs and analysed using spreadsheets.

It is noted that other languages, such as Java, offer similar facility but C++ was favoured because of its widespread use in real-time industrial control system applications. Although the main purpose of the software code produced in this research was for investigation and demonstration, it is hoped that it will also provide a basis for a real-world prototype agent system at some later date, in which case, the choice of C/C++ should ease the task of translation without restricting the choice of target control hardware.

5.2 Program Organisation

The simulation program comprises three main parts. These are the multi-agent system (MAS), the model of the fluid flow environment with which the agents interact and the user interface. The principal data flow involved is shown in the diagram below.

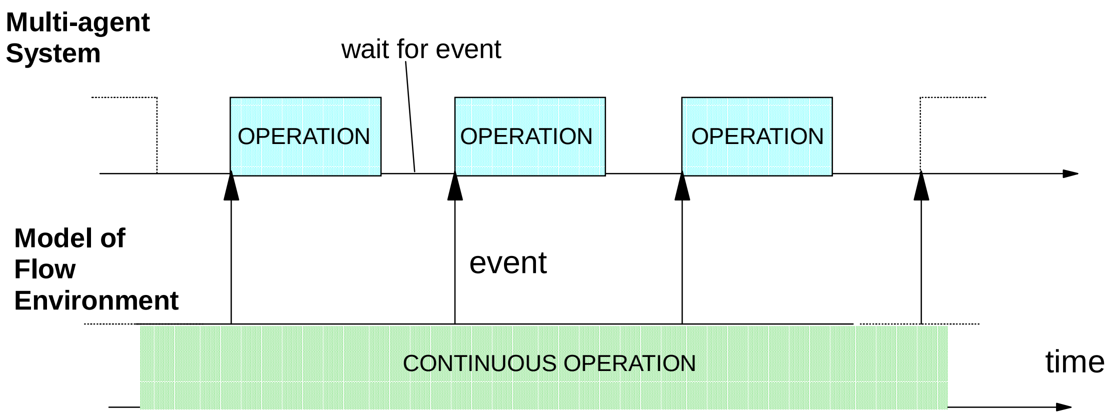
Fig 5.1 Simulation Program Organisation



The flow model is a single software entity, i.e. program module, whereas the multi-agent system comprises a set of entities representing agents. To simulate real-world behaviour the models must run concurrently and this was achieved by use of program threads.

In the single processor environment of the PC, threads are managed by the Windows multi-tasking operating system. From the point of view of simulation there may be no need to synchronise model threads. However, practical experience shows that if the cycle times of two continuous (apparent) concurrent processes differ significantly, then the operating system favours the faster process and overall program speed degrades. This undesirable situation is avoided if the two threads are synchronised by introducing a wait state into the faster of the two processes. During the wait state, the processor is fully available for the slower process with the result that overall program speed is increased. This arrangement is shown diagrammatically below.

Fig 5.2 Thread Synchronisation



Thread synchronisation was incorporated into the simulation program design from the outset. However it is emphasised that this interaction between the flow model and the agents is quite distinct from any behavioural interaction which is being modelled in the simulation.

The method of data exchange between the models depends on the communication mode adopted in the agent system design and how faithfully this is modelled in the program. At the simplest level, practical experience confirms that data can be reliably transferred between program threads by means of global variables. This method was used for transfer of program control variables.

The detail of the user interface is determined by the specifics of the particular design phase. In general, the interface was chosen for simplicity to be a single maximised window which simultaneously displays the overall performance of the IGC and the status and behaviour of the multi-agent system. Graphical representation was used wherever possible for clarity and effect. Simple pull-down menus and some of the ‘common controls’ available within the Windows API, e.g. track bars, up-down controls were used to facilitate user input for controlling the simulation.

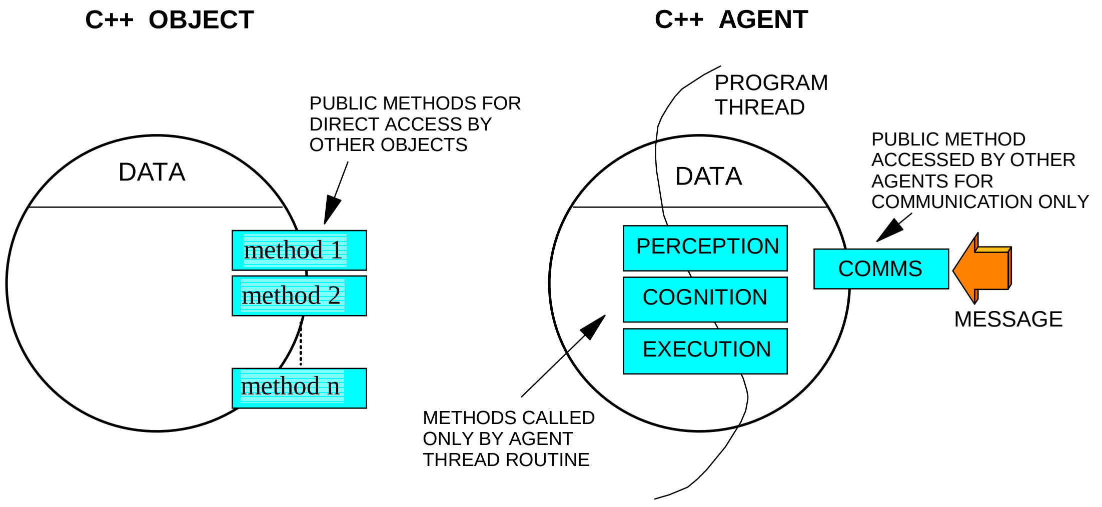
5.3 Agent Representation

There is no facility in the C++ program language which enables an agent to be represented directly so it was necessary to define a software entity, in terms of the available program constructs, which would serve this purpose. Specifically, an entity was required which controls the choice and timing of its own actions and, in this way, appear to behave as an autonomous agent.

The highest level of abstraction available in C++ is the object and this enables a software entity to be defined which encapsulates data and methods. But the methods defined within an object class are 'public' i.e. they are available for execution by other objects or modules. In other words, an object, unlike an autonomous agent, does not determine what particular action it takes or when this action occurs. However, if an object is instantiated in its own program thread and if the object class methods are only called by the associated thread routine then the behaviour of the object becomes self-determining. Admittedly, there is no construct within the language to ensure that this condition is met so this really relies upon programming discipline. An exception arises in the case of agent communication, where it was decided to assign a public method which could be accessed directly by other agents for purposes of passing messages. (This is similar in concept to the method of distributed object communication supported by the CORBA standard described in chapter 3).

Thus the general form of agent representation adopted in this research may be summarised in the following diagram which also emphasises the essential differences between an 'agent' and an 'object'.

Fig 5.3 Agent Representation



It is noted that Ferber (1999) makes the same distinction stating that 'objects encapsulate data and methods, whereas agents encapsulate behaviour'.

CHAPTER 6

IGC Design Phase 1

Single Agent and Global Control of Stator Rows

6.1 Objectives

This phase of the research was essentially one of 'problem analysis' in which the overall objective was to determine the design requirements for the multi-agent system which performs the control function of the intelligent geometry compressor.

To determine the functions of the MAS and how these may be achieved requires an understanding of the behaviour of the fluid flow environment with which the MAS interacts; in particular, of the relationship between the variable geometry manipulated by the MAS and the performance parameters of the machine which are to be controlled. The first objective was therefore to develop a suitable model of the flow system and incorporate this into a computer program thus enabling the operation of the machine to be simulated over a range of operating conditions.

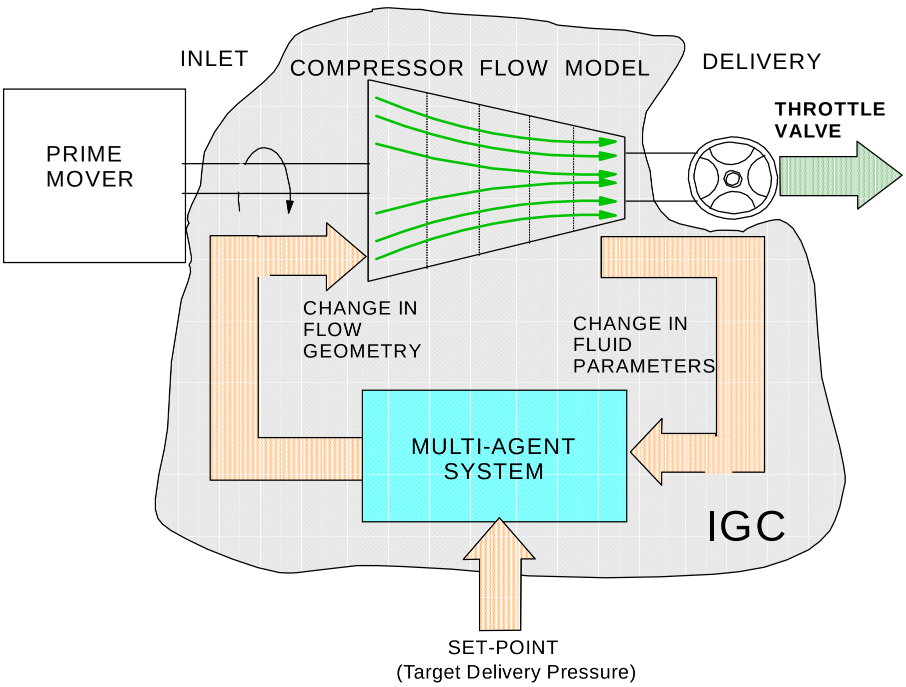
The objectives of simulation trials were to capture the operating characteristics of a variable geometry compressor and to reveal the potential for enhancing the machine performance through agent control. The former was pursued by parametric variation in 'open loop' operation of the compressor model whilst the latter was investigated by introducing a simplified MAS, effectively a single agent, capable of providing 'closed loop' control by means of global adjustment of stator rows. In this way a 'datum' performance was established for later comparison with the multi-agent system developed in the next phase of the research.

6.2 Total System Model

A diagram of the total system model is shown in Fig 6.1 in which the main constituent parts are separated for convenience of representation. External to the IGC is the downstream fluid load represented by a single variable throttle valve that was sized so that the full flow range of the IGC could be investigated. A linear law approximated the relationship between flow rate and pressure drop across the throttle. The prime mover was assumed to be ideal and thus able to provide constant rotational speed irrespective of torque variation at the output shaft. Neither the throttle valve nor the prime mover is subject to automatic control in this system but their settings can be changed to allow the operating point of the IGC to be varied.

The IGC model comprises the compressor flow model and the MAS. At this level of abstraction, attention was focused on system behaviour and there is no explicit representation of internal components such as vanes, sensors and actuators.

Fig 6.1 Total System Model



The flow model and MAS are described in detail below.

6.3 Flow Model

In order to develop a quantitative representation of IGC performance and of the relationships between flow geometry and fluid parameters, the flow model needs to be based on a particular configuration of axial flow compressor.

Usually, the physical form of a fixed geometry axial flow compressor, in terms of number of stages, dimensions and blade geometry, is designed to achieve a required operating point corresponding to a particular combination of delivery pressure and mass flow rate at a specified rotational speed. This arrangement is termed the 'nominal' specification of the machine and the corresponding operating point the 'design' point. For the IGC, a configuration of 5 stages and an inlet guide vane was chosen, as a compromise between realistic representation of a multi-stage axial compressor application and the computational effort required for simulation. The stagger angles of all stators are variable. This degree of variability in the flow model enables a wide range of MAS control options to be investigated. (Note: for ease of description the term 'stator' is used generally in this text to mean 'a row of stator vanes' including the inlet guide vane row).

There has been a great deal of research over many decades into methods for predicting the performance of multi-stage axial flow compressors. Mostly, the methods are based on analytical models of an underlying flow regime e.g. 1- or 2-dimensional flow combined with correlation data obtained from practical experiment on real machines. Examples are to be found in Howell and Calvert (1978), Wright and Miller (1991) and Camp and Horlock (1993). A computer code for predicting axial compressor performance is described by Steinke (1982). The methods reported in the literature are generally aimed at producing accurate predictions of compressor performance under specific operating conditions including off-design and, as a result, often involve complex calculation to take account of 3-dimensional flow effects, radial variation in blade form and blade tip effects.

For present purposes, such refinements in the flow model are not necessary. It is sufficient that the model is able to generate operating characteristics, which are similar in general form, to those associated with axial flow compressors and to respond to changes in flow geometry in a consistent manner. It is also desirable that the chosen method for modelling compressor flow is relatively easy to code and involves limited computational effort.

The prediction method of Howell and Bonham (1950) was selected because it meets these requirements and has widespread acceptance in the literature.

The Howell and Bonham method predicts the characteristics of a single axial compressor stage based on 2-dimensional incompressible steady flow analysis combined with correlation data from practical experiment. The method is reproduced in a number of standard works in the field such as Horlock (1958) and Shepherd (1956). For purposes of this research it was necessary to extend the method to allow for variable stator angle and also to combine (or stack) stage results to achieve overall characteristics for a 5-stage configuration. This extension to the analysis provides for variation in fluid density from stage to stage and therefore introduces some allowance for compressibility effects into the model. It should be stressed that this representation strictly describes steady-state flow i.e. fluid parameters are not dependant on time. The model is therefore suitable for investigating the effects of variable geometry on the steady-state performance of the machine but not dynamic behaviour. Full details of the analysis and calculation method are given in the Appendix.

The data for a real compressor stage used by Howell and Bonham to validate their prediction method was used here in order to lend realism to the flow model. Five such stages were combined to provide the overall machine configuration. Scaling the stage diameters in relation to local fluid density preserved similarity of performance of each stage. Using this data and the calculation method described in the Appendix enabled the nominal specification of the machine to be determined. This is given in the table of Fig 6.2, which may be interpreted as the specification of the (hypothetical) IGC when the internal flow geometry is set by the MAS to achieve the nominal design point.

It should be noted that the overall dimensions of the IGC are comparable to those of axial flow compressors used in large-scale industrial applications of the sort described in Chapter 2. This was done deliberately in order to keep the number of stator vanes in the machine to a manageable level in anticipation of investigating the control of individual vanes at a later point in the research.

Fig 6.2 IGC Nominal Specification

COMPRESSOR STAGE CONFIGURATION AND IDENTIFICATION	STAGE GEOMETRY: BLADE OUTLET ANGLE	INLET FLOW ANGLES	OUTLET FLOW ANGLES	ROTOR DIAMETER RATIO	PERFORMANCE AT DESIGN POINT: FLUID DENSITY RATIO	EFFICIENCY %	FLOW FUNCTION	PRESS. RISE COEFF.	STATIC PRESSURE AT OUTLET bara	STAGE REACTION %
	β_0	$\alpha_1 = \alpha_3$	$\alpha_0=\alpha_2=\alpha_4$	D_R	$\overline{\rho}$	η	φ_x	ψ	p	R
VARIABLE INLET GUIDE VANE (IGV)	5.6	-	13.1	1.0000	1.000	-	0.837	-	0.977	-
Rotor **STAGE 1**		43.9	13.1	1.0000	0.975	90.2	0.837	1.024	1.617	50
Variable Stator	5.6		13.1							
Rotor **STAGE 2**		43.9	13.1	0.8947	1.393	90.2	0.837	1.024	2.349	50
Variable Stator	5.6		13.1							
Rotor **STAGE 3**		43.9	13.1	0.8250	1.816	90.2	0.837	1.024	3.161	50
Variable Stator	5.6		13.1							
Rotor **STAGE 4**		43.9	13.1	0.7740	2.241	90.2	0.837	1.024	4.042	50
Variable Stator	5.6		13.1							
Rotor **STAGE 5**		43.9	13.1	0.7347	2.668	90.2	0.837	1.024	4.990	50
Variable Stator	5.6		13.1							
OVERALL PERFORMANCE AT DESIGN						89.9	0.837	6.019	4.990	-

Note: Blade and flow angles are in degrees

CONSTANTS:

p_0 = 1.013 bara inlet static pressure

ρ_0 = 1.225 kg/ m3 inlet gas density

D_0 = 1.03 m rotor outer diameter at inlet stage

D_H = 0.3 m rotor hub diameter at inlet stage

N = 6000 rpm rotational speed

λ = 0.93 work done factor

s/c = 0.742 blade pitch-chord ratio

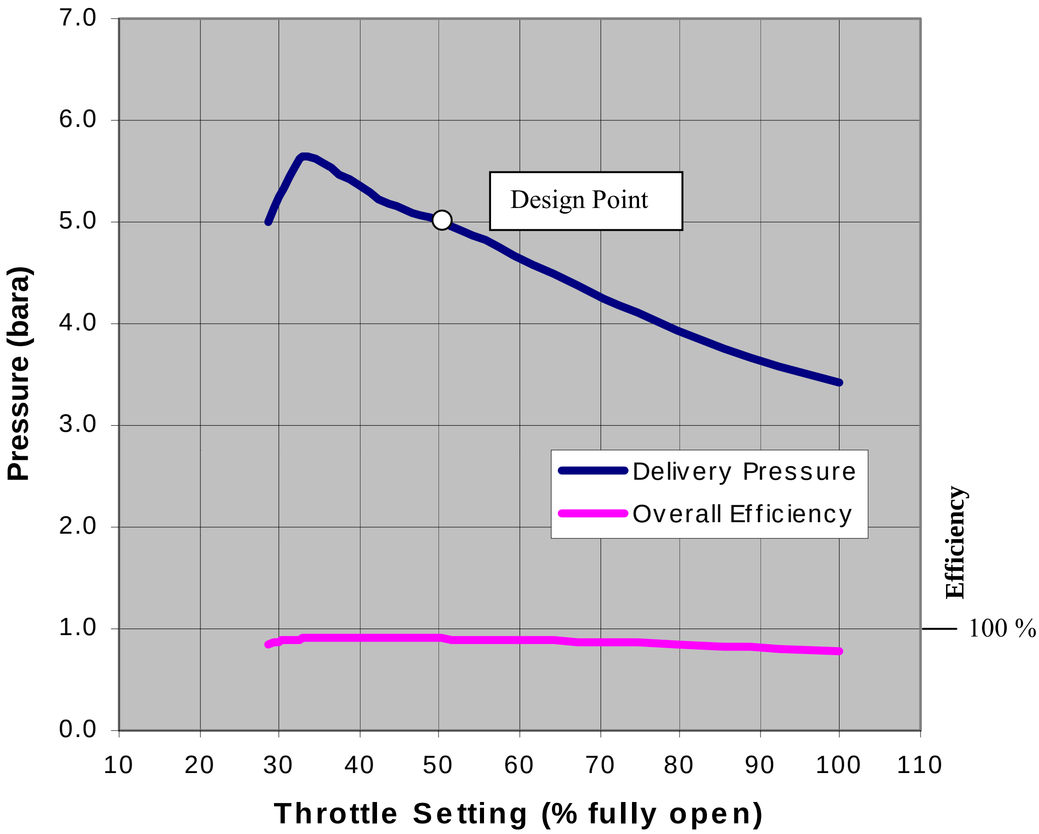
h/c = 2.0 blade aspect ratio

IGV and stator blade outlet angles are variable over range -35 to +35 degrees

(Note that, in this model, *stator vane outlet angle* and *stagger angle* are synonymous.)

For the nominal or 'design' setting of the stators the compressor steady state performance is represented by graphical characteristics of delivery pressure and efficiency. These characteristics are usually given against flow rate but here, the abscissa is chosen to be throttle valve setting as shown in Fig 6.3.

Fig 6.3 Delivery Characteristics for IGC at Design



As the throttle setting is changed the operating point follows the locus of the characteristic and delivery pressure varies accordingly. The limitation on minimum flow, in practice, is set by the onset of stall conditions. Maximum flow is limited by choked conditions or stall, either of which may result in some form of flow instability. Whatever the exact mechanics of the resulting breakdown in flow regime, the consequential effects are potentially damaging and therefore operation beyond these flow limits is to be avoided. For fixed geometry machines, i.e. stators not variable, the design operating point is selected to provide an adequate margin of operational stability - usually referred to as the stall or surge margin, as explained in Chapter 2.

The form of the compressor characteristic, and hence the location of the point at which stall occurs, reflects, largely, the pressure loss across blade rows due to aerodynamic drag. Such losses are directly related to the direction and magnitude of the inlet flow velocity vector to each blade row, which in turn varies with throttle setting. The direction of the velocity vector relative to blade angle, or ***incidence***, is critically related to blade drag coefficient, as explained in the detailed analysis given in the Appendix. Thus, for purposes of simulation, incidence was adopted as the defining parameter for stable operation in steady state and used to determine a ***stall margin*** parameter for each row of blades in the machine. (Note: in this research 'stall margin' should strictly be interpreted as 'static stall margin'.)

Howell (1945) proposed that onset of stall occurs when the pressure loss across a compressor cascade (blade row) is approximately twice the minimum loss. For present purposes this condition was adopted as the definition of the point at which the stall margin for a blade row is zero. For a given stage, the stall margin was taken as the lesser of the stall margins for the stator and rotor respectively. Similarly, the overall stall margin for the compressor was taken as the minimum of all stage stall margins. For this research the correctness of definition of stall margin is not important, since it was required only that the flow model provides some limiting condition, representative of stall, which could be used in the context of multi-agent control.

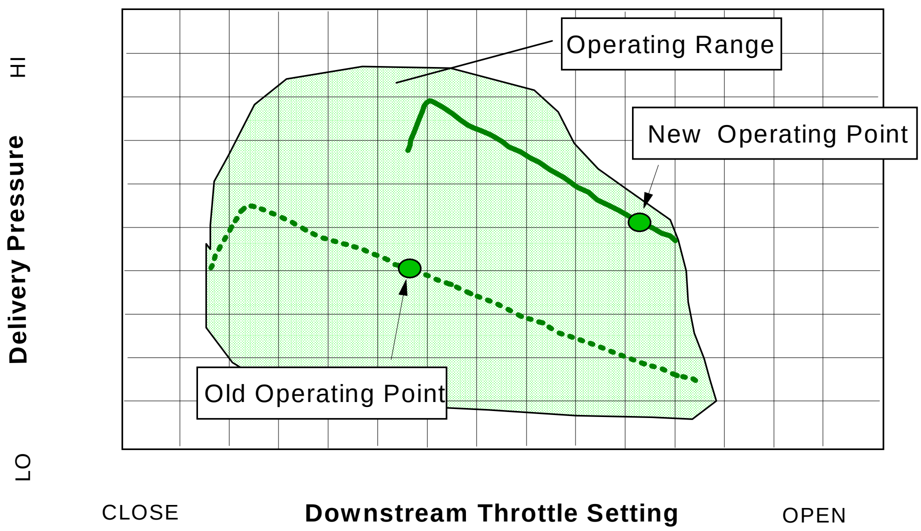
The model allows stator angles, and hence the internal flow geometry of the machine, to be varied and this enables a wide range of performance characteristics to be generated. The objective of automatic control of stator angles is to produce enhanced machine performance compared to an equivalent fixed geometry machine. This may be achieved in terms of extended operating range, regulation of delivery pressure or mass flow and maximising machine efficiency.

6.4 MAS Design

At this juncture, the relationships between the settings of individual stator rows and flow parameters were unknown and so the MAS was conceived as a single agent which applies global adjustment of stators in order to achieve regulation of delivery pressure. Regulation specifically means the achievement and maintenance of delivery pressure at a required level, or set point, regardless of throttle setting. Global adjustment of stators means that vanes within all stator rows are adjusted equally from their design settings.

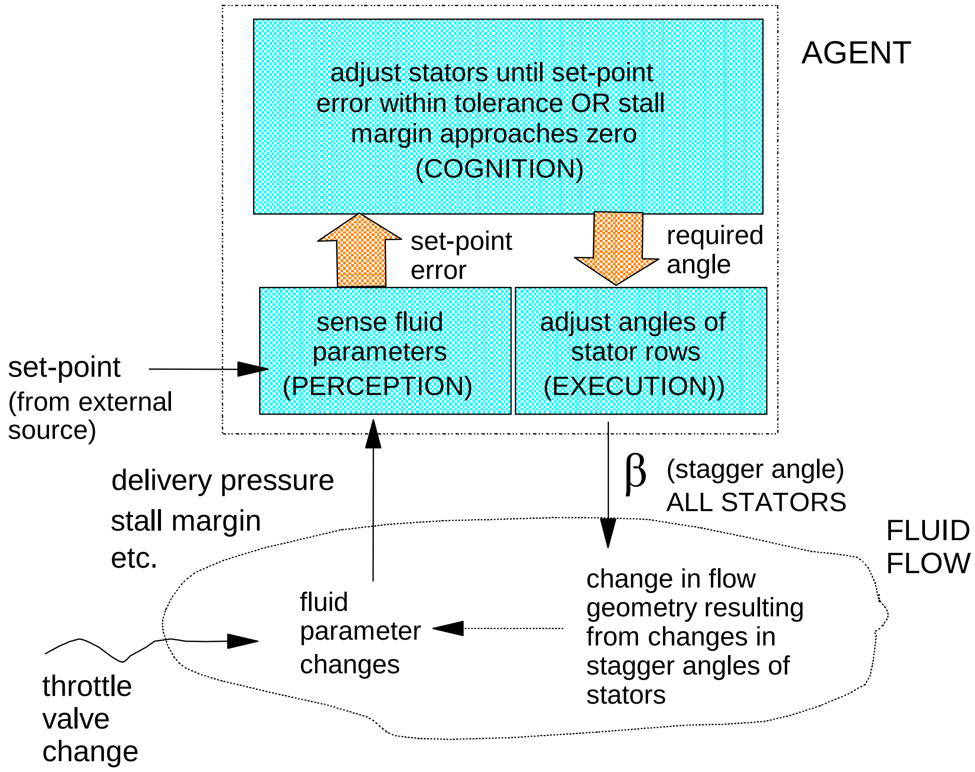
The operating range of the compressor is defined here as the area of the steady-state performance map within which the overall stall margin is greater than zero at all points. The boundary of the operating range is thus determined by the static stall limits referred to earlier. Ideally, the machine should be controlled to operate only within this boundary as illustrated in Fig 6.4.

Fig 6.4 Regulation of Delivery Pressure



The agent design was based on the generic architecture noted in Chapter 3, which comprises modules for perception, cognition and execution. This nomenclature was retained for consistency, even though it is not correct to regard the simple control action as cognitive behaviour. To achieve regulation of the compressor delivery pressure the agent behaves as a simple closed-loop controller as illustrated in the diagram below.

Fig 6.5 Single Agent Architecture and Operation



Initially, the control strategy was to apply a simple proportional action to eliminate the delivery pressure error. Later, following simulation trials, the control strategy was developed to include methods for limiting stall and maximising efficiency. It was assumed that all fluid parameters which are subject to control are able to be sensed and values are available to the agent as required.

6.5 Simulation Program

The simulation program implements the flow model, defined by the IGC nominal specification, and the MAS control function. In its initial form, the simulation program was designed to support the following main tasks:

a) validate the flow model

b) investigate relationships between variable geometry and fluid flow parameters

c) demonstrate regulation of delivery pressure by single agent

The overall structure of the program comprises three modules, as described in the previous chapter, and each module comprises one or more C code source files. The main features of the program are described in the following sections.

6.5.1 Main Program and User Interface (Module: *IGC1main*)

The main program and the related Windows function support the user interface and calls the C functions which run the flow model and the MAS respectively. As previously indicated, the user interface is presented in a single maximised window and provides a number of dedicated controls for running the simulation and displays results in both graphical and numerical form.

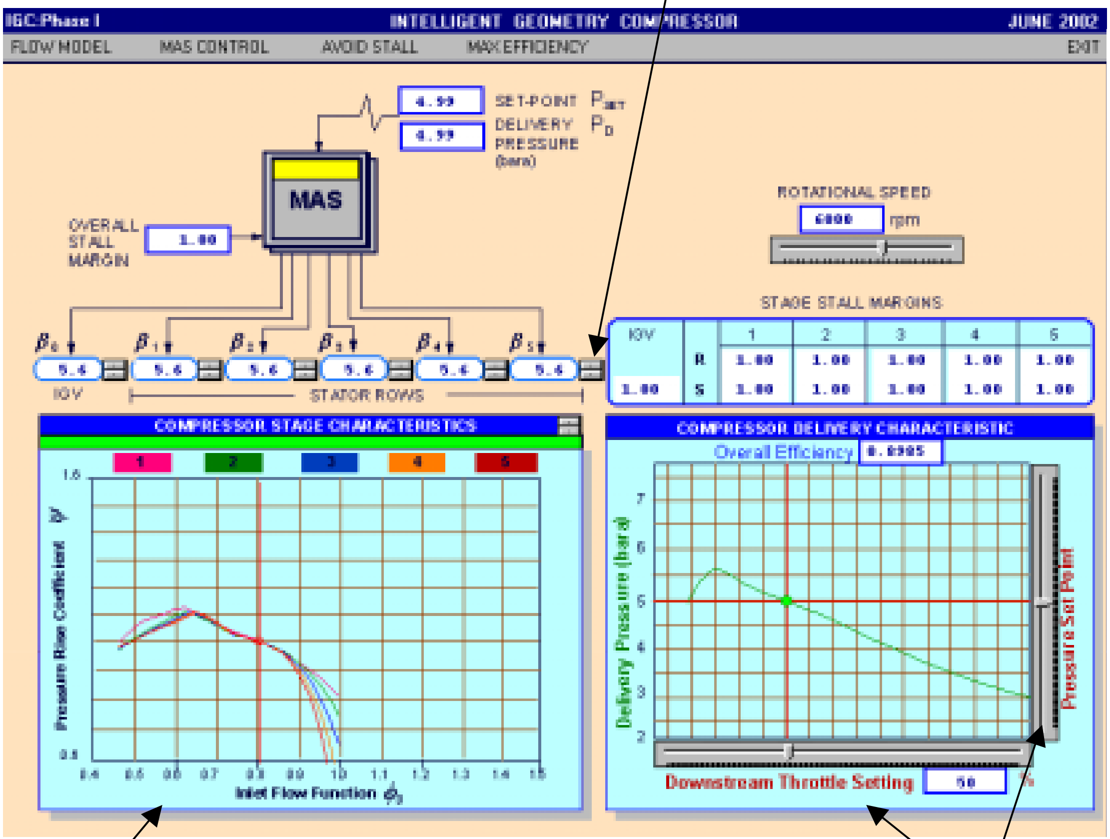
The general form and operation of the main program and the Windows function follow a fairly standard approach that can be found in introductory texts on Windows 95 programming such as Schildt (1995). The user controls and the program outputs are indicated in the screen image of the simulation window shown in Fig 6.6. Facilities for writing simulation results to data files are included in the main program. These are activated as required by changes to source code rather than through the user interface.

6.5.2 Flow Model (Module: *Runflow*)

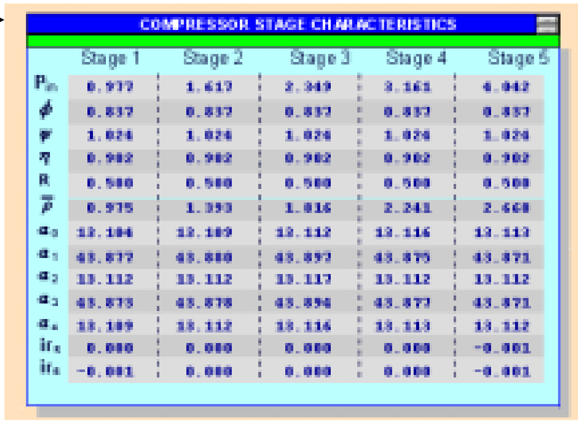
This module is called by the main program in response to menu selection. It initiates a program thread within which the flow model calculation is repeatedly performed until interrupted by the user. At the end of each computation cycle, an 'event' flag is set (for the reasons given in Chapter 5) and the output data in the simulation window is refreshed. The code is written in procedural form (as opposed to object-oriented) and, for simplicity, is decomposed into a number of functions. The operation of the module is illustrated in the simplified flowcharts shown in Fig 6.7.

Fig 6.6 Phase 1 Simulation Window

User controls to adjust setting of individual stator rows



User-selectable displays

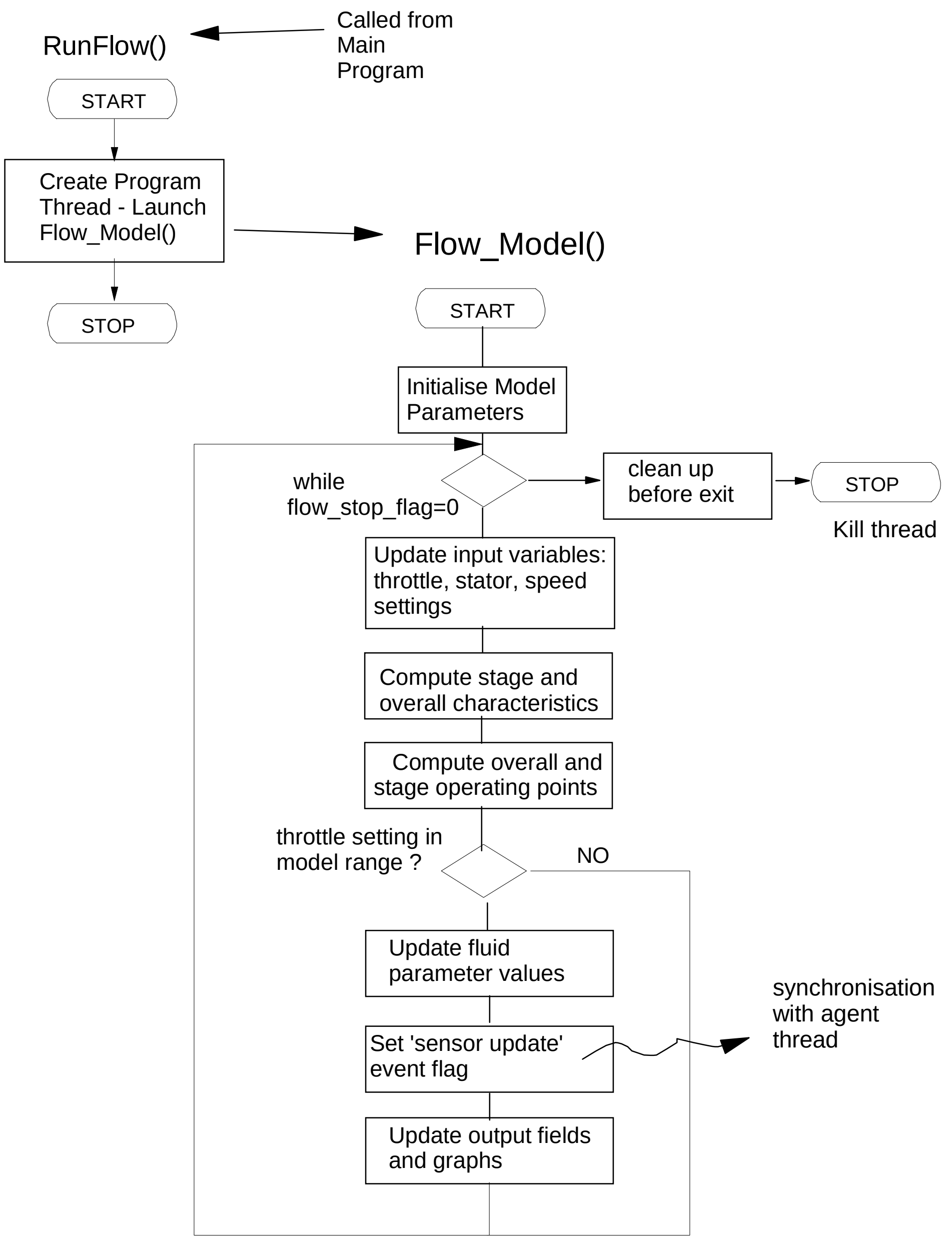


COMPRESSOR STAGE CHARACTERISTICS

	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5
P_{in}	0.977	1.613	2.349	3.161	4.042
ϕ	0.837	0.837	0.837	0.837	0.837
ψ	1.024	1.024	1.024	1.024	1.024
η	0.902	0.902	0.902	0.902	0.902
R	0.500	0.500	0.500	0.500	0.500
$\bar{p}$	0.975	1.393	1.816	2.241	2.668
α_0	13.104	13.109	13.112	13.116	13.113
α_1	43.877	43.880	43.893	43.875	43.871
α_2	13.112	13.112	13.117	13.112	13.112
α_3	43.873	43.878	43.894	43.877	43.871
α_4	13.109	13.112	13.116	13.113	13.112
ir_R	0.000	0.000	0.000	0.000	-0.001
ir_S	-0.001	0.000	0.000	0.000	-0.001

User controls for operating point

Fig 6.7 Flow Model Operation (Simplified Flow Charts)



On each computation cycle the model re-calculates the stage and overall characteristics using the procedure described in the Appendix. The characteristics are held in arrays. The intercept between the compressor characteristic and the linear throttle characteristic is determined by linear interpolation. This locates the overall operating point i.e. pressure rise coefficient and inlet flow function. The latter is then used to locate the stage operating points, again by means of interpolation of the stage characteristic data. At the end of the cycle, selected fluid parameter values are made available to the MAS via global variables. Also, before starting a new computation cycle, the output displays and data fields are updated. Principally, these are the stage characteristics, a table of flow parameter values for each stage, the overall compressor characteristics and a ‘stall status’ table which displays the current stall margin values for all stator and rotor rows. A stall margin of zero corresponds to a value of incidence ratio (defined in the Appendix) of ! 0.4 according to the definitions adopted in section 6.2 above, whereas a stall margin of unity indicates that the incidence for the row is the same as at design.

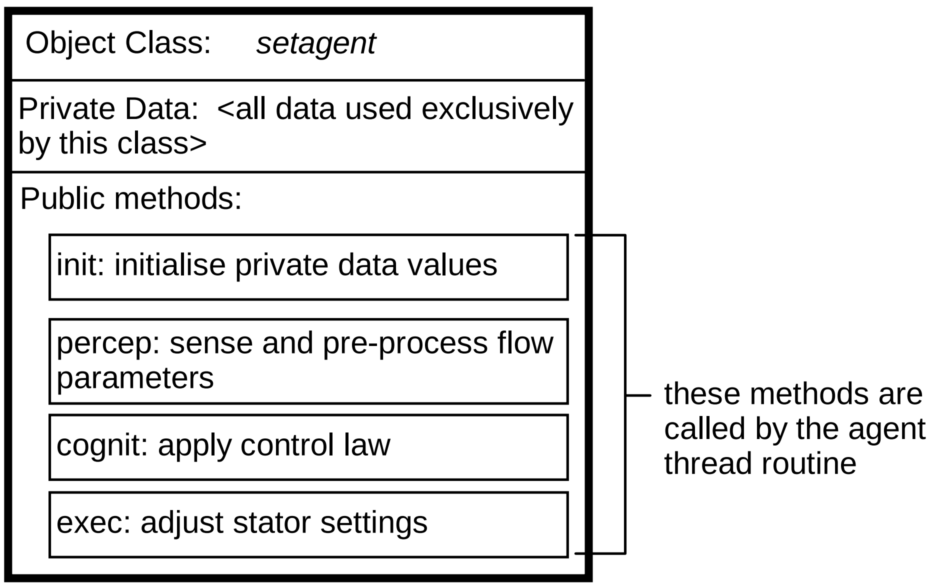
Limits are imposed on the flow model operation in two respects. Firstly the stage characteristics are calculated over a flow range within limits of incidence ratio of -0.6 to +0.6. These limits represent the extent of Howell's correlation data. Should a flow value give rise to a value of incidence ratio outside these limits then the calculation aborts and the model proceeds with the next flow value. The stage and overall compressor characteristics displayed in the simulation window reflect the range of flow values for which the incidence ratio is within the ! 0.6 limits. Thus the occurrence of out of range flow values will truncate the appearance of the displayed characteristics.

The second limit imposed by the flow model concerns the overall operating point. If the flow rate resulting from a particular throttle setting is outside the range of flow for which the overall compressor characteristic has been calculated then no update of the model takes place. This condition is simply beyond the scope of validity of the analytical model. It will be noted from Fig 6.7 that when this limit applies then no synchronisation flag is set and therefore the agent remains inactive in a ‘wait’ state. As a result the displays on the simulation window appear to ‘freeze’. However, normal operation can usually be recovered by resetting the user controls for throttle and set point to an operating point, which is within the valid range.

6.5.3 MAS implementation

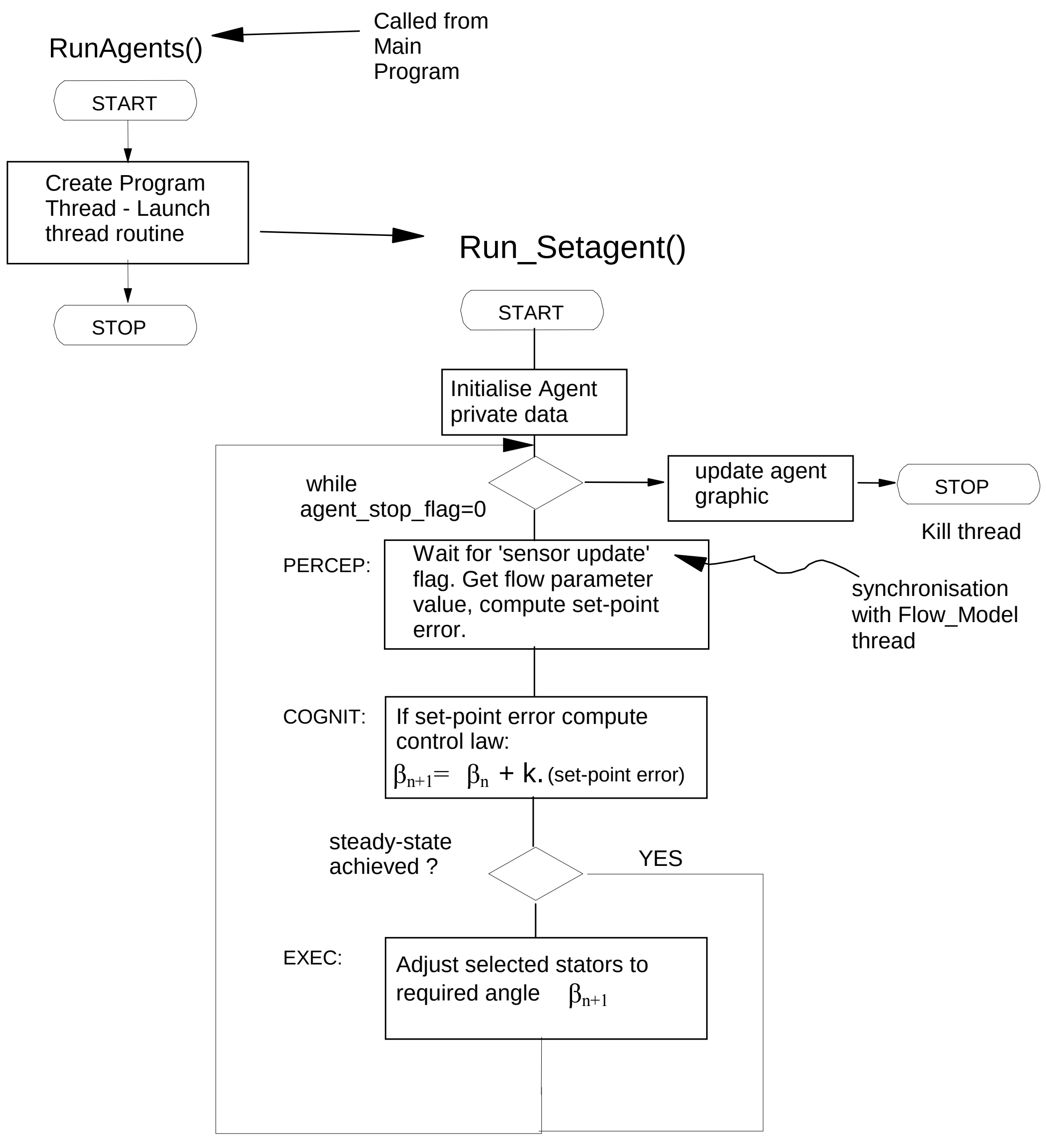
The single agent representing the MAS was implemented in the program code as a C++ object running in its own thread following the convention for agent representation defined in the previous chapter. The object class is named *setagent* (set-point agent) and comprises methods for initialisation, perception, cognition and execution. In this case the methods are very simple routines.

Fig 6.8 Object Class Definition for Single Agent



When agent control is selected by the user, the main program calls a function RunAgents() which simply creates a new program thread and launches a thread routine named *run_setagent*. This routine instantiates the object named *setagent* of class setagent within it and initialises the private data associated with the object. The agent is thus activated and cycles continuously until de-selected by the user. On each cycle, the perception, cognition and execution modules are executed sequentially. In operation, the agent effectively runs concurrently with the flow model and the respective program threads are synchronised as previously described in Chapter 5.

Fig 6.9 Agent Operation (Simplified flow charts)



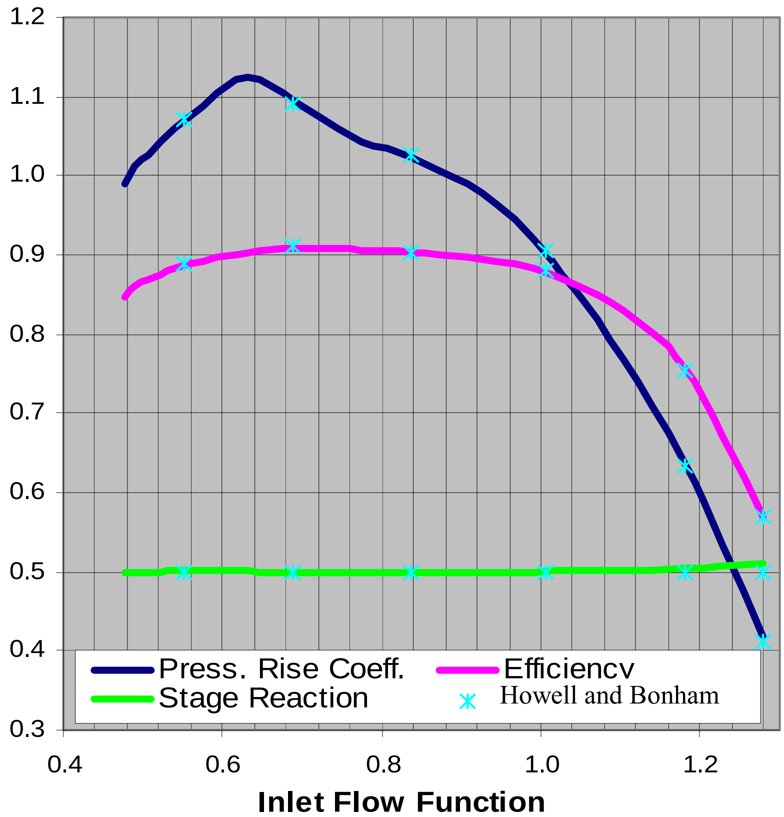
6.6 Simulation Trials

Simulation runs were firstly conducted in ‘open-loop’ mode i.e. with the agent inactive in order to obtain basic data about compressor performance and behaviour. In subsequent trials the agent was activated in order to evaluate closed-loop control.

6.6.1 Validation of flow model

The first objective was to capture stage characteristics at design and off-design in order to confirm the correctness of implementation of the flow model. The operation of the downstream stage, stage 5, most closely approximates that of an isolated stage and was therefore the one chosen for comparison with the published data of Howell and Bonham (1950). The results are shown below.

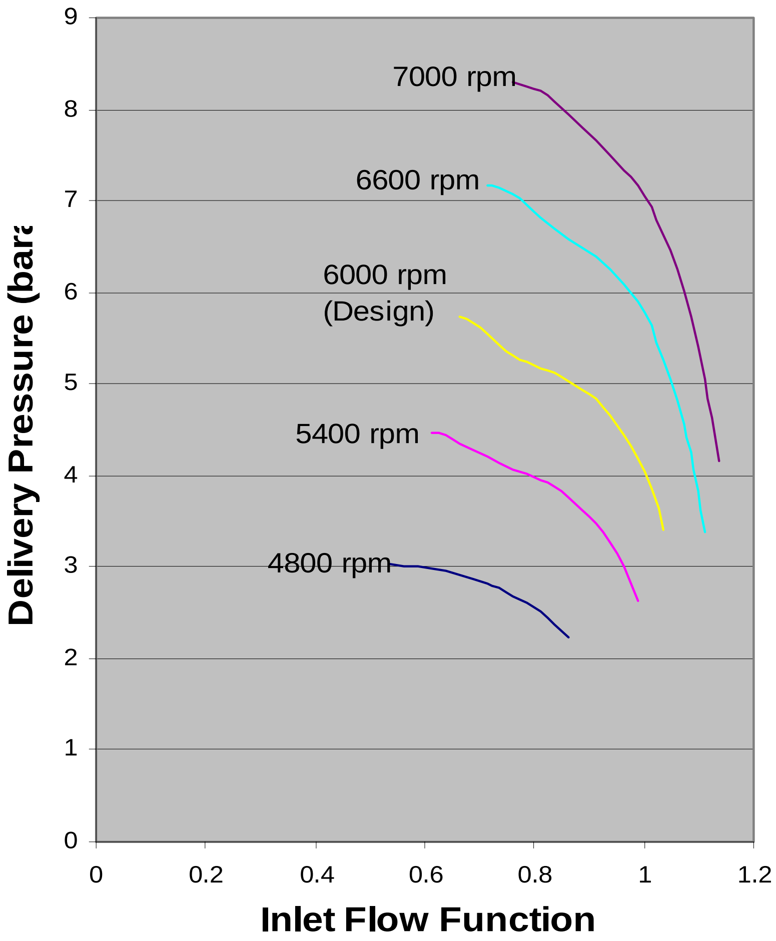
Fig 6.10 Stage 5 Characteristics at Design



The coincidence of computed values with those of Howell and Bonham indicated that the calculation routine developed in the Appendix had been implemented correctly.

The overall pressure-flow characteristics for different rotational speeds were then captured. Of interest is the general form of these results in comparison with those often presented in the literature.

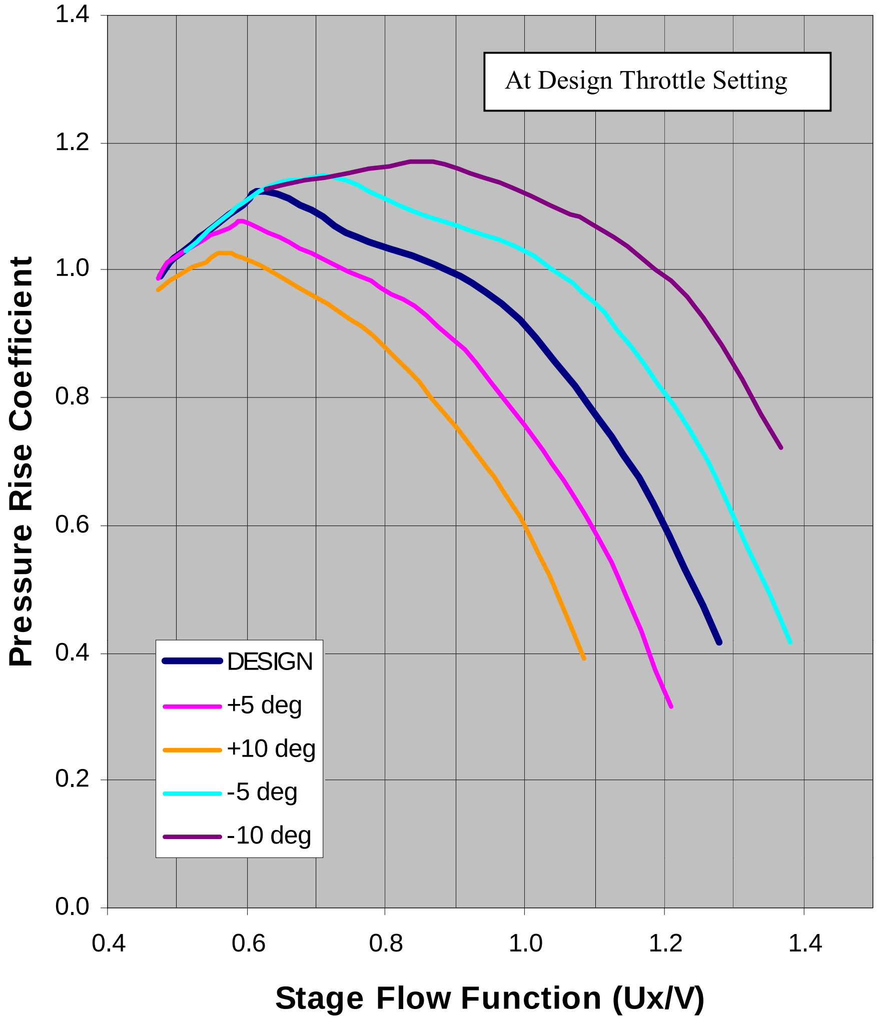
Fig 6.11 Overall Characteristics at Various Speeds



There are clear similarities between the computed results and, for example, the typical characteristics shown in Fig 2.3 of Chapter 2 taken from Gresh (1991). In particular, the characteristics become steeper and shift upwards and to the right as speed increases. This qualitative comparison suggests that the flow model correctly represents the effects of rotational speed on the performance of the machine.

The third basis for validation of the flow model was to consider how a stage characteristic changes as a result of change in angle of the upstream stator. Again stage 5 results were used for this purpose.

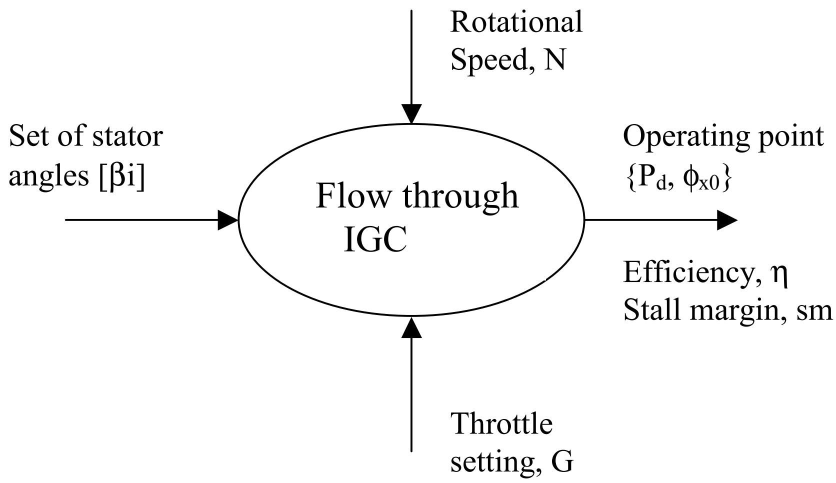
Fig 6.12 Effect of Stator Setting on Downstream Stage Characteristic
(Computed results for Stage 5)



A qualitative comparison of the above results with the published data given by McCoy and Hooper (1953) and Steinke (1982) shows general agreement in both the sense and relative magnitude of displacement of the characteristics with change in stator angle.

6.6.2 Overall Parametric Relationships

Any overall operating point of the compressor is uniquely defined by the tuple of delivery pressure and inlet flow function $\{P_d, \phi_{x0}\}$. Associated with an operating point are values of overall efficiency and overall stall margin. In addition, the operating point is dependent on the internal flow geometry, the throttle setting and the rotational speed as illustrated in the sketch below.



For this research, only operation at design speed was considered so of primary interest were the relationships between stator angles and the operating point variables at given throttle settings.

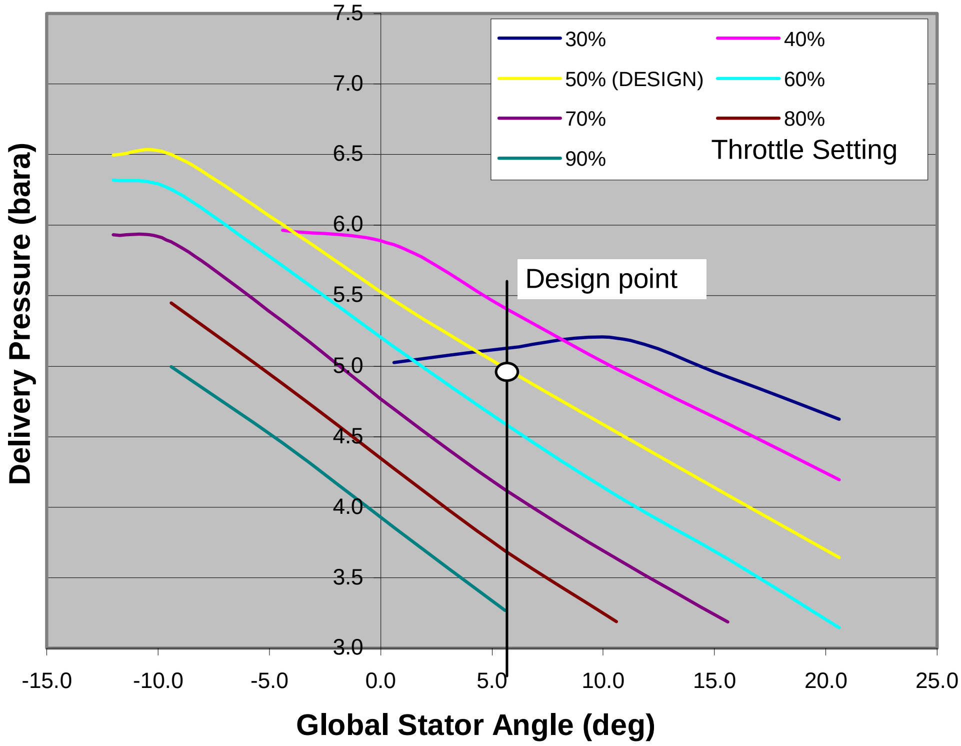
In this phase of the research, to simplify matters, the case was considered whereby all stator angles are equal so that a single global variable β is sufficient to define the internal geometry rather than a set of variables, $[\beta_i]$. In this way the overall machine operation may be likened to that of a single variable stage. The limitation on range of β is set by the model limits on incidence ratio described earlier. For reasons given in the Appendix, the incidence at the IGV is dependent only on β and not flow velocity. Given the model limits then the range of β for the IGV was calculated to be about ± 18 degrees from design i.e. -12 to + 24 degrees approximately. These then are the limits on global stator angle.

A temporary facility was provided in the simulation program to cause all stator angles to be adjusted equally under manual control. This enabled open-loop characteristics of the main performance parameters of interest against global stator angle to be acquired over a range of throttle settings. The results are described below.

a) Delivery Pressure

The relationship between delivery pressure and global stator angle was found to be generally linear over a wide range of throttle settings as shown in Fig 6.13. The slope of the linear regions is negative and corresponds to operation on the negative slope of the delivery pressure-throttle characteristic (ref. Fig 6.3). The non-linearity arises as the operating point effectively moves onto the positive slope of this characteristic and pressure then reduces with reducing flow. At throttle settings less than design, i.e. < 50%, the operating point is closer to the peak of the delivery pressure vs. throttle characteristic, and therefore a relatively smaller change in stator angle from design is required before pressure begins to decrease.

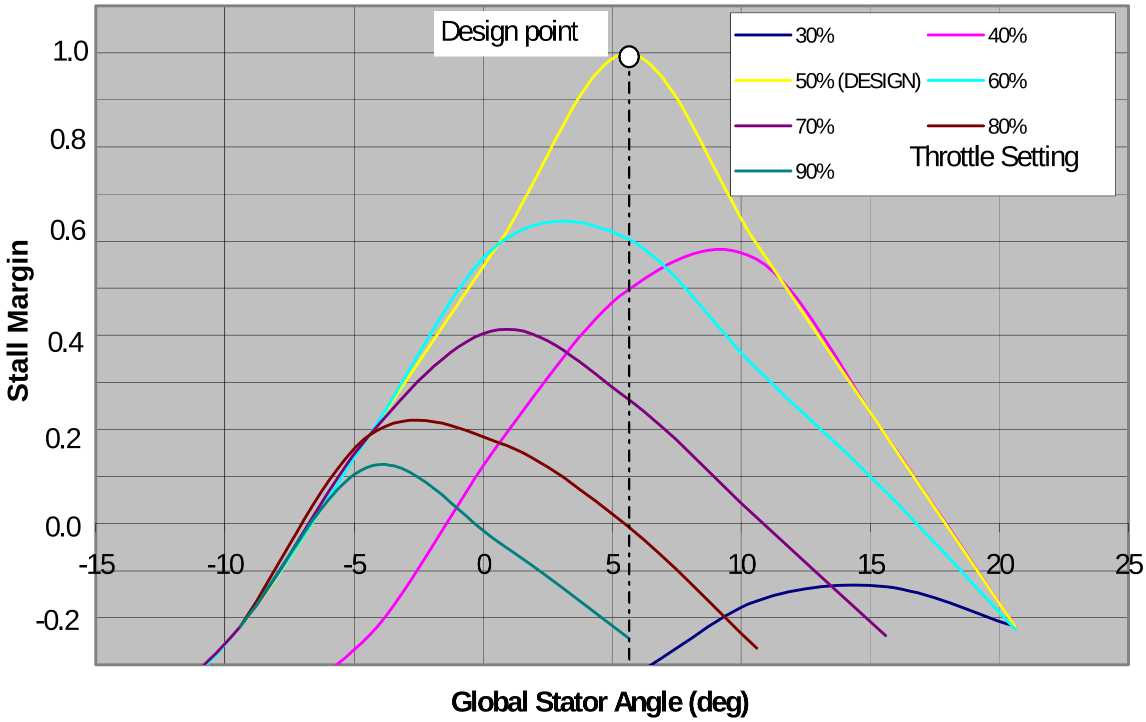
Fig 6.13 Delivery Pressure v Global Stator Angle



b) Overall Stall Margin

The graphs of overall stall margin vs. stator angle shown in Fig 6.14 indicate that a maximum value of stall margin exists for any given throttle setting. This maximum value reduces as the throttle setting departs further from the design point. Similarly, the range of stator angle for which the stall margin is greater than zero diminishes at off-design throttle settings. It was noted that the minimum throttle setting for positive stall margin is about 32%.

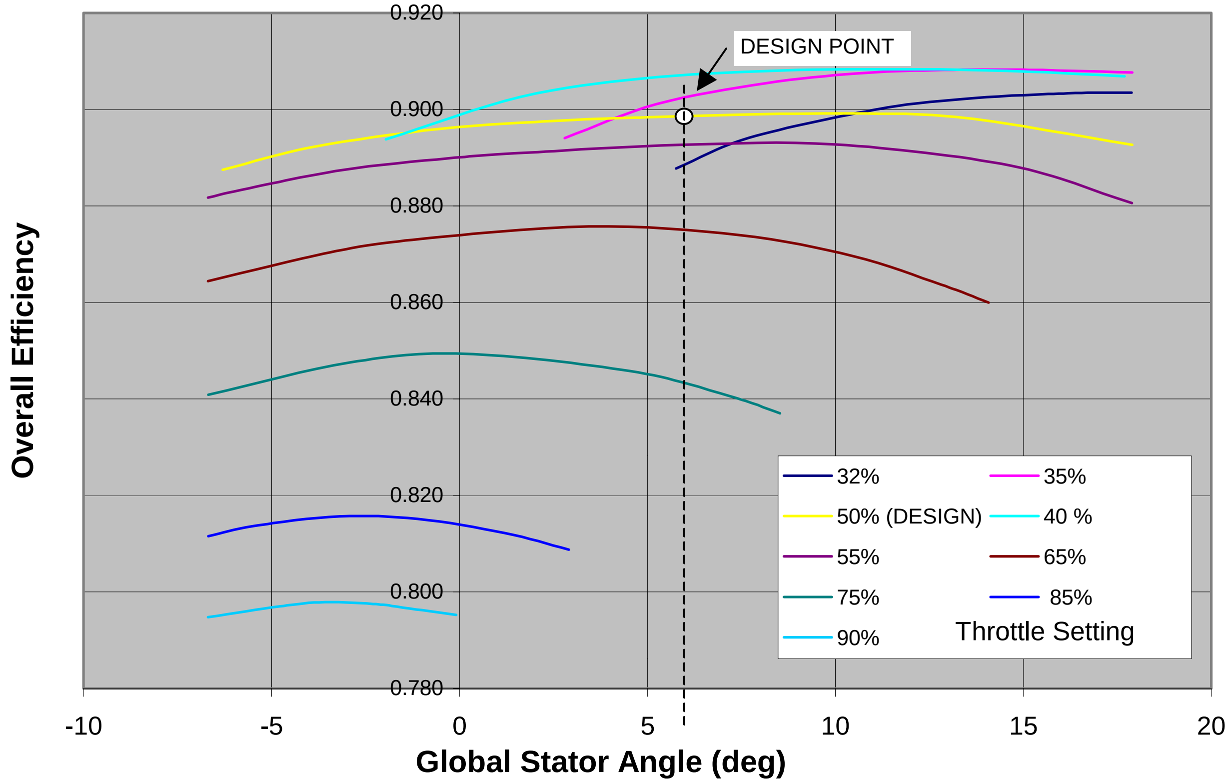
Fig 6.14 Overall Stall Margin v Global Stator Angle



c) Overall Efficiency

As with stall margin, efficiency has a maximum value at a given throttle setting although the total variation is relatively small as shown in Fig 6.15. Of interest is the difference between the maximum value and that at design. Generally, this difference is insignificant except at throttle settings well away from the design setting of 50%. At these extremities the difference may approach 2% (abs. value) which is a significant value. For example, Sun and Elder (1998) report an efficiency improvement of 2.68% as a result of a numerical optimisation of stator setting in a multi-stage axial compressor. For a large machine, such apparently small increases in efficiency could mean substantial savings in operating costs.

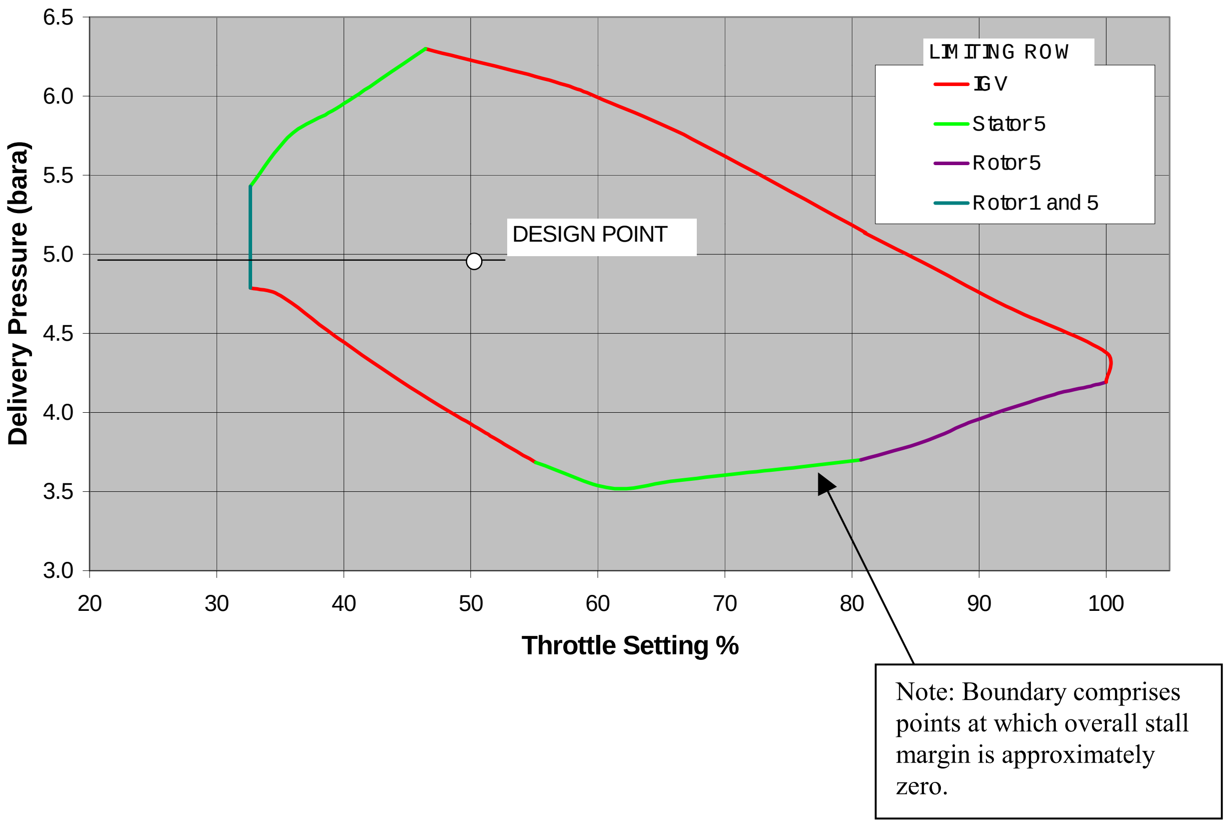
Fig 6.15 Overall Efficiency v Global Stator Angle



d) Operating Range

Traversing the delivery pressure-throttle field, using global stator adjustment and capturing values of pressure and throttle setting at which the overall stall margin is approximately zero, enables the boundary of operation of the machine to be defined. The result is shown in Fig 6.16. As previously explained, overall stall margin is determined by the blade row, rotor or stator, which has the smallest stall margin. Thus segments of the operating boundary correspond to particular rows for which the stall margin is near zero. This is also indicated in Fig 6.16. which shows that the greater part of the boundary relates to stall at the inlet guide vane (IGV). The minimum throttle setting is 32%, as noted above, since at lesser values the overall stall margin is less than zero. The upper limit on throttle setting is close to 100%, i.e. throttle fully open.

Fig 6.16 Compressor Operating Range



6.6.3 MAS Control

The remainder of the simulation trials in this phase were conducted with the single agent active and providing closed loop control.

a) Delivery Pressure Regulation

The object of regulation is to achieve and maintain a target delivery pressure, or set point, regardless of change in throttle setting. The control is a simple proportional action as described earlier in section 6.4.

The control law is a simple linear rule of the form:

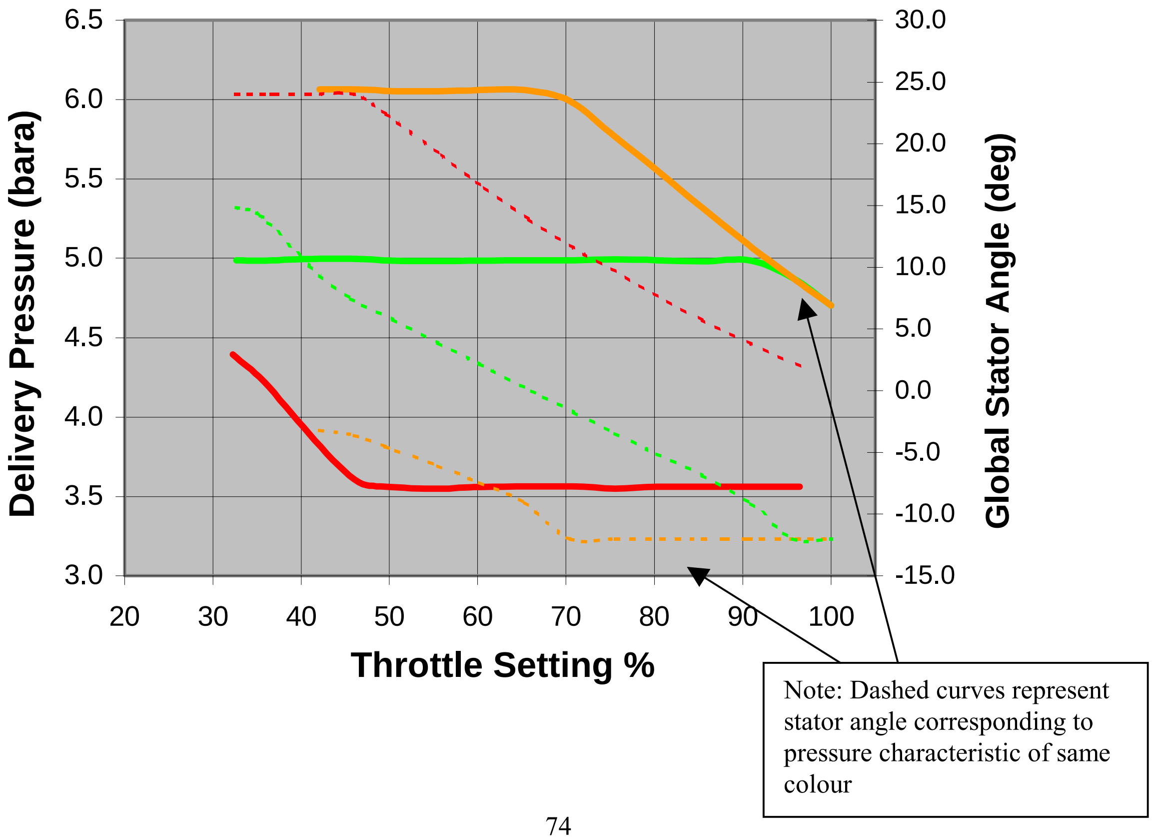
$$\beta_{n+1} = \beta_n - k.(P_{SET} - P_D)_n$$

where k = gain constant and n refers to last data update

The set-point value, $P_{SET,\ is}$ input to the simulation by means of a trackbar control and the agent updates the current value of delivery pressure, P_D , on each cycle of operation.

Sample regulation characteristics are shown in Fig 6.17. These indicate the steady-state performance of the control action, which is effective in maintaining a given set point over a range of throttle settings constrained only by the limits of the model. For example, the design pressure level of approximately 5 bara is maintained from 32% to just over 90% throttle setting. As noted earlier, for global stator control the operating limits are ultimately set by the IGV angle i.e. -12 to +24 degrees. It will be noted on Fig 6.17 that pressure regulation ceases when these limits are reached.

Fig 6.17 Pressure Regulation Characteristics



The effectiveness of this simple control with constant gain is not surprising given the linear relationship between delivery pressure and global stator angle noted earlier. Since the model is essentially kinematic i.e. does not embody any dynamic, time-dependant, elements then response is determined entirely by timing constraints imposed within the simulation program.

b) Limiting Stall Margin

It is clearly desirable to constrain the control system from adopting operating points at which the overall stall margin is less than zero i.e. points which lie outside the operating range as defined in Fig 6.16. It is possible to introduce constraints on set-point change to avoid this situation but it is not possible, strictly, to prevent excursion outside the operating range due to a change in throttle setting. Therefore any strategy for stall avoidance must also be accompanied by a facility for 'stall correction' which will limit the extent and duration of such excursion. In practice, given that delivery pressure changes in such a large machine, due to downstream load disturbances, are unlikely to be rapid coupled with the ability to set the operating boundary condition to include some 'safety factor', then a corrective action strategy seems feasible.

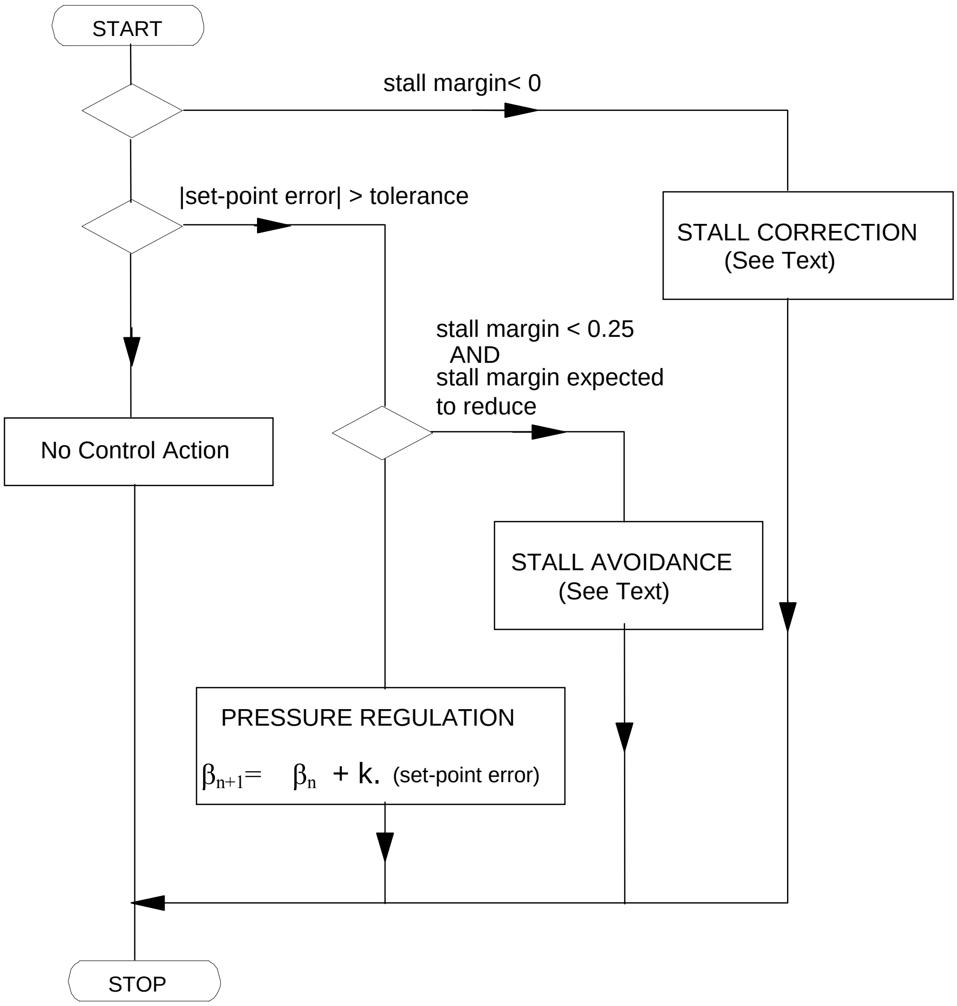
The control algorithm developed for limiting stall margin is shown in outline in the flowchart of Fig 6.18. The main additional elements are the routines for 'stall avoidance' and 'stall correction'. These are described in detail below.

i) Stall Avoidance

Before reacting to a delivery pressure error, the agent needs to determine how the resulting action may affect the overall stall margin and to modify its response accordingly. The approach adopted is based on the following simple analysis, which determines the sense (i.e. arithmetic sign) of the 'next' change in stall margin.

Fig 6.18 Control Algorithm for Limiting Stall Margin

(Simplified flow chart)



Stall margin is a function of global stator angle and throttle setting hence:-

$$sm = f\{\beta, G\}$$

and for a small change dβ at constant throttle setting the corresponding change in stall margin is approximated by:-

$$dsm = \left(\frac{\partial sm}{\partial \beta}\right). d\beta$$

Denoting 'last' values of variables by the subscript n and 'next' values by the subscript n+1, then the next value of stall margin change is given by:-

$$dsm_{n+1} = \left(\frac{\partial sm}{\partial \beta}\right).d\beta_{n+1}$$

The partial derivative represents the slope of a stall margin vs global stator angle characteristic of the type shown in Fig 6.14. The form of the characteristic has a single maximum point and, for present purposes, it is sufficient to know on which side of this maximum the current operating point lies. Thus only the sense of the partial derivative and not the magnitude is required to be computed.

The sense of the partial derivative can be found from the product of the arithmetic signs of the last changes in the constituent variables i.e. sign(dsm_n) and sign($d\beta_n$). The sense of the next change in stator angle is determined by the sign of the pressure error since:-

$$d\beta_{n+1} = -k.(P_{SET} - P_D)_n$$

then $\quad sign(d\beta_{n+1}) = -sign((P_{SET} - P_D)_n)$

From the above the sense of the next change in stall margin is obtained as the product of signs of last changes in variables:-

$$sign(dsm_{n+1}) = -sign(dsm_n) \,.\, sign(d\beta_n) \,.\, sign((P_{SET} - P_D)_n)$$

It should be noted that the changes dsm_n and $d\beta_n$ are computed in the same interval of time and within which there has been little or no change in throttle setting i.e. dG~0 in order to comply with the definition of the partial derivative.

The above result is used to determine the action taken by the agent when the stall margin is positive and the pressure error is outside tolerance. If sign(dsm_{n+1})>0 then the next stator adjustment will cause the operating point to move towards the set-point and away from the operating range boundary, in which case the original pressure control law may be safely applied.

If sign(dsm_{n+1})<0 then a modified form of the control law is applied of the form:

$$\beta_{n+1} = \beta_n - k.(P_{SET} - P_D)_n .sm_n$$

In this case the magnitude of the stator adjustment is conditioned by the prevailing value of stall margin whilst the sense of adjustment is consistent with reducing set-point error. The modified control law is effectively a 'product of errors' and will drive stator adjustment until either the set-point error or the stall margin error (i.e. relative to zero) is eliminated. In this way the operating point will be constrained to remain within the desired operating range. In practice, the modified control law is only applied if the operating point is relatively close to the boundary of the operating range i.e. stall margin <0.25.

ii) Stall Correction

The first priority of the agent cognition module is to test if stall margin is less than zero and, if so, to apply corrective action. Again a simple proportional control law is adopted using the stall margin as the 'error'. The sense of adjustment is derived from the preceding analysis so the control law becomes:

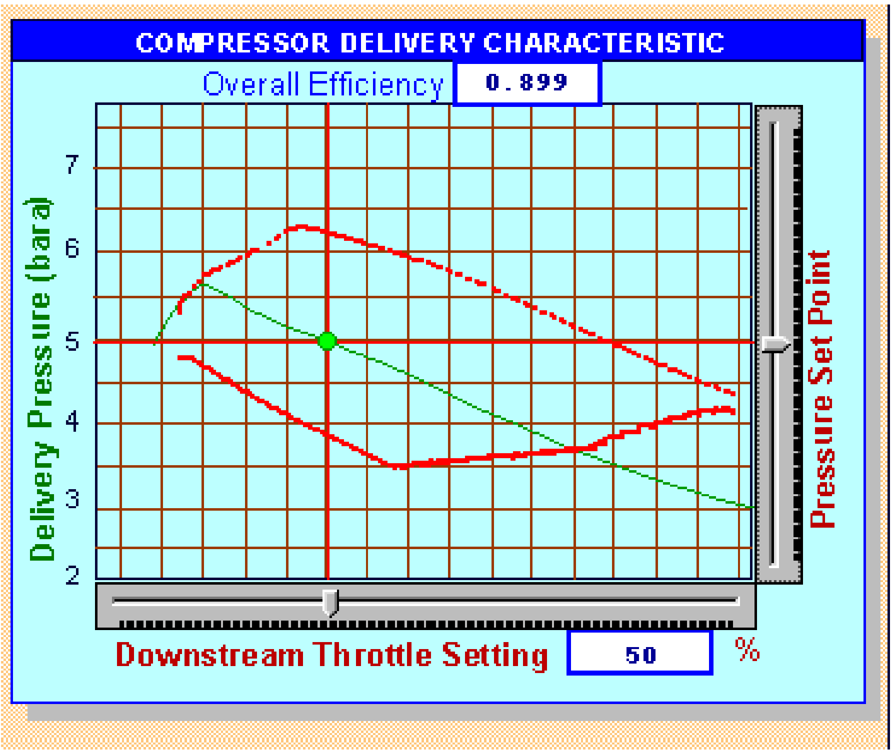
$$\beta_{n+1} = \beta_n - \text{sign}(dsm_n) . \text{sign}(d\beta_n) . (k.sm_n - \delta)$$

Since stall margin, sm_n , is negative then this law always causes the sense of stator adjustment to be in the direction of increasing stall margin. The inclusion of the small constant δ provides 'momentum' to the calculation, ensuring that a positive value of stall margin is achieved.

The routines for stall avoidance and correction were incorporated into the agent code of the simulation program and are invoked by selecting the 'AVOID STALL' entry on the main menu bar following the selection of 'MAS CONTROL'.

In addition, a feature was added so that operating points at which the overall stall margin is close to zero (actually < 0.025) are marked on the screen display thus highlighting the boundary of the operating range. The result of a simulation run is shown in the screen image of Fig 6.19. and was obtained by successively traversing the full range of throttle and pressure settings. The red contour thus represents the boundary of the steady-state operating range obtained by agent control and compares very closely with that obtained manually and shown in Fig 6.16.

Fig 6.19 Agent-Controlled Operating Range



The operating range was limited by the minimum throttle setting of 32% imposed by the flow model as previously explained otherwise agent control was successful in confining steady-state machine operation within the indicated boundary. As expected, temporary excursions outside the boundary occurred when throttle changes were made at points already on or very close to the boundary and in a direction, which caused stall margin to decrease. In such cases the corrective action applied by the agent quickly restores the operating point to the boundary.

c) Maximising Efficiency

An alternative control objective to pressure regulation is to find the point of maximum overall efficiency at a given throttle setting. The basic principle of the method adopted is firstly to determine the required direction of change in global stator angle necessary to increase efficiency and then to continue adjustment until no further increase is detected. A new search is triggered whenever a change in throttle setting occurs.

Since it is desirable to retain the control on stall margin described in the previous section, the method of maximising efficiency actually adjusts pressure set-point rather than stator angle directly. In this way the system behaves exactly as in pressure regulation mode except that set-point values are generated from within the routine for maximising efficiency rather than being input by the user. In addition, to accelerate the search for the maximum point, the target set-point value is determined from a rule of the following form:

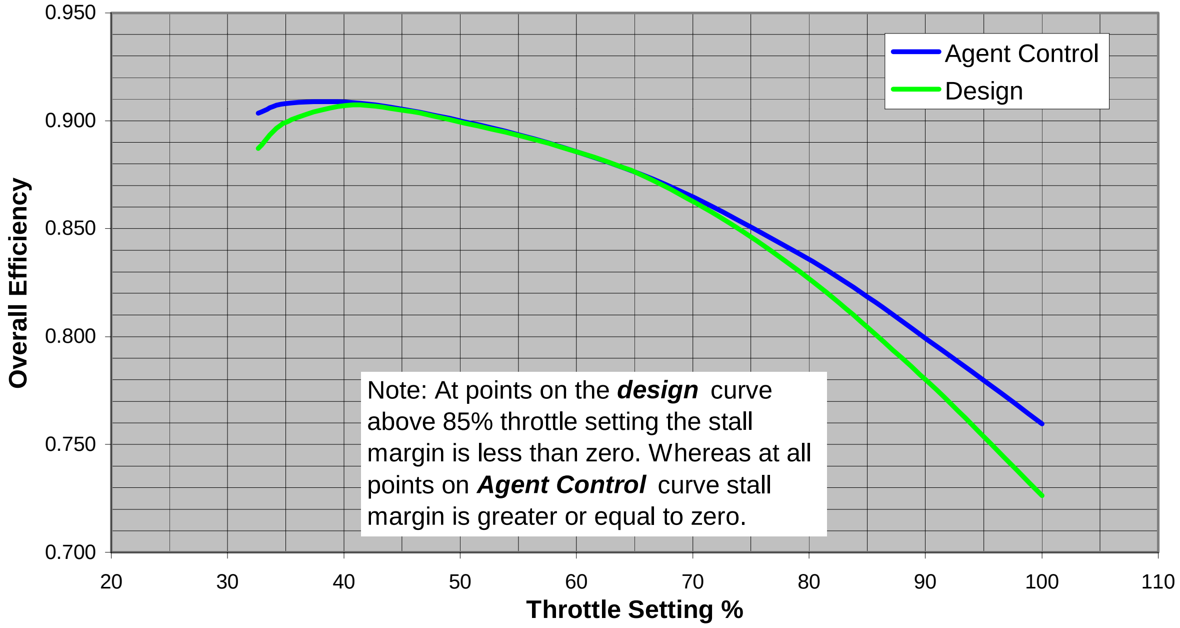
$$P_{SET} = P_D + \mathrm{dir}\eta \, . \, k \, . \, |(\eta_{max} - \eta)|$$

where $\mathrm{dir}\eta$ = sign of required change as found from initial trial

At the start of a search the value of η_{max} is set to 1.0 and then, at the end of the search, η_{max} is set to the actual value found. In this way the pressure error term in the original control law is effectively replaced by an 'efficiency error' term.

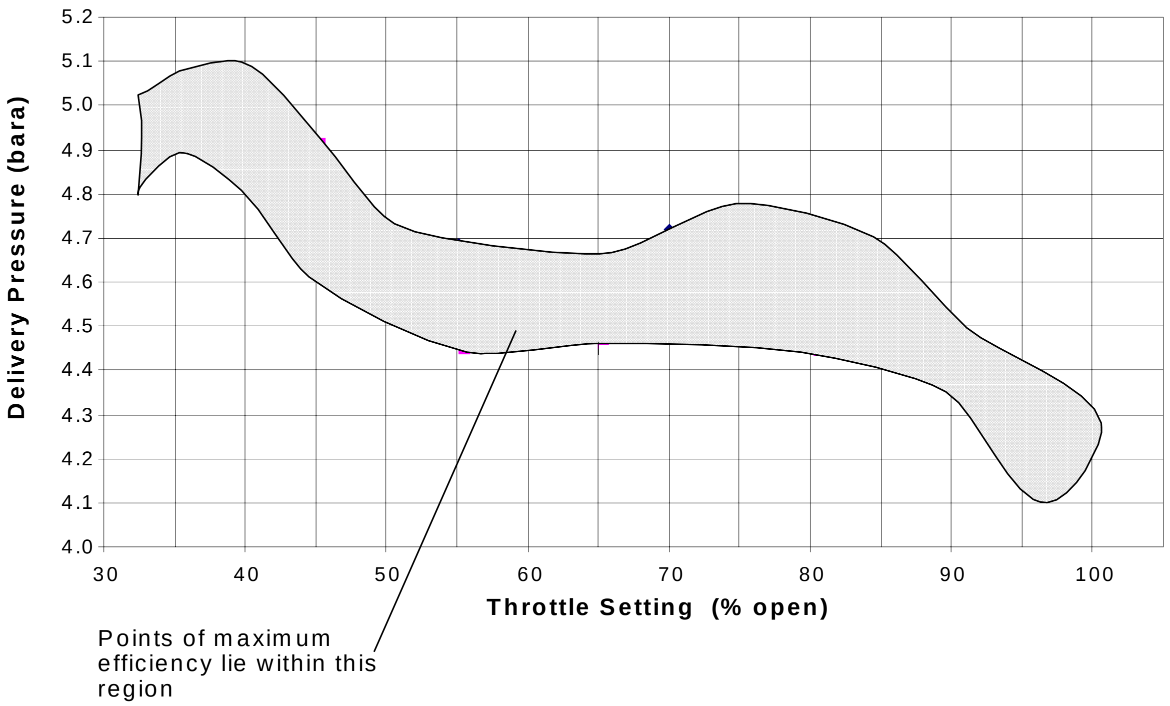
A menu selection 'MAX. EFFICIENCY' was added to the simulation program and is invoked after selection of 'MAS CONTROL' and 'AVOID STALL'. The results of simulation runs with and without the efficiency control are shown in Fig 6.20. As expected from earlier observations (ref. Fig 6.15), differences between maximum values and the corresponding values at design are insignificant except at the extremes of throttle setting. Nonetheless, the control algorithm was shown to be effective and the results coincided exactly with those obtained manually.

Fig 6.20 Control of Efficiency



Also, because the efficiency characteristics are relatively 'flat' then no change in the maximum value is apparent over a range of pressure values at a given throttle setting. This effect is illustrated in Fig 6.21 below.

Fig 6.21 Delivery Pressure under Efficiency Control



6.7 Conclusions

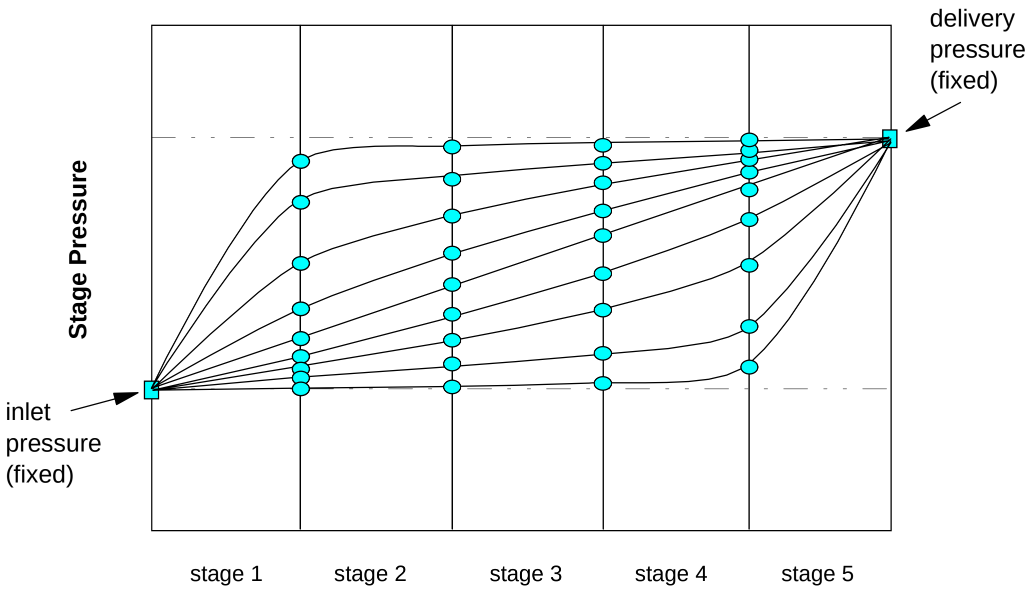
In this first phase of work, a nominal specification for the hypothetical IGC was established and the effects of variable geometry on performance studied with the aid of computer simulation. From the results and observations of the study, the requirements for the proposed multi-agent system were clarified and some initial insight into how these may be met was gained.

Firstly, a reference condition for machine control was defined in which a single agent acting as a simple closed-loop controller adjusts all variable stator rows equally. This condition provides a datum performance for delivery pressure regulation, the extent of which is bounded by the onset of stall conditions at one or more of the blade rows. For present purposes, stall was defined in terms of a limiting value of incidence angle relative to the value at design for a given blade row. The ability to maximise overall efficiency through global stator control was also demonstrated.

The functional objective for the multi-agent system is to achieve IGC performance that exceeds that of the datum control system. Scope for improved performance exists if variable stators are adjusted independently rather than globally. This was evident, for example, when considering the operating range of the machine under global stator control. The extent of the operating range is determined by the occurrence of stall at any one of the blade rows even though the other rows may be well away from the stall condition. Ideally, if all blade rows achieve zero stall margin simultaneously then the operating range is bound to be extended.

It was not clear at this juncture if independent control of stators would enable higher values of efficiency to be achieved in comparison with those achieved under global stator control. However, it appeared possible to maximise efficiency at a particular operating point rather than at just a particular throttle setting so that efficiency and pressure control could be complementary rather than alternative objectives. This is supported by the recognition that, in theory, a given delivery pressure is achievable by any one of a large number of internal stage pressure combinations. This point is illustrated in the sketch below.

Fig 6.22 Theoretical Internal Pressure Distribution along Axis of Machine



It would be expected that stage efficiency would change, to some extent, with change in stage pressure rise. And since overall efficiency is determined ultimately by the combined effect of stage efficiencies then for each internal pressure distribution a different value of overall efficiency would result. Each of the internal pressure distribution characteristics relates to a particular combination of stator row angles and throttle setting, therefore it should be possible to achieve different values of overall efficiency by individual stator adjustment whilst maintaining delivery pressure. The underlying physical explanation is that stator adjustment effectively alters the distribution of pressure losses within the machine and thus the values of individual stage efficiencies.

CHAPTER 7

IGC Design Phase 2

Multiple Agents and Independent Control of Stator Rows

7.1 Objectives

The IGC performance requirements were identified in the previous phase of work as pressure regulation, stall avoidance and maximising overall efficiency. It was also shown that the operating range of the machine could be increased if stator rows are independently controlled. The objective for this phase of the research was therefore to design a multi-agent system with independent stator control and to demonstrate the resulting improvement in overall machine performance in comparison with the datum control system, previously defined, which is able only to adjust stators globally.

The approach to the conceptual design of the multi-agent system was derived from the reference material presented in chapter 3. The objective was to arrive at an initial system design, which could be implemented in code and incorporated into a new version of the simulation program. Following the practice of the previous design phase, the results of simulation runs were then used to further develop agent methods and improve overall performance of the system.

7.2 Total System Model

The total system model described in the previous chapter and depicted in Fig 6.1 remained valid for this phase of work. As before, there is no explicit representation of real system components such as sensors and actuators. However, it is clear from the previous work that certain sensors are implicit in the system model if the information necessary for control is to exist and be accessible to agents. The information required includes the following main control variables:-

- delivery pressure
- stall margin (as previously defined) for each stator and rotor row
- stagger (blade outlet) angles for each stator row
- overall efficiency

It was recognised that information such as stall margin and efficiency, in practice, would not be directly available from simple sensors but would need to be derived from processed data generated by a combination of other, possibly intelligent devices. The definition of the sensor system is thus a significant subject in its own right and is beyond the scope of this research. For present purposes it was simply assumed that all information about the fluid environment required by agents to fulfil their respective tasks is available.

7.3 Flow Model

A model of steady flow through a 5-stage axial compressor with variable stator rows was developed in the previous phase of work and incorporated into the simulation program, IGC1. The nominal specification of this hypothetical machine is given in Fig 6.2. No changes to this model were necessary to enable the objectives of this phase of work to be achieved and therefore the related computer code was carried over directly into the modified simulation program, IGC2.

7.4 MAS Design

The understanding of multi-agent systems gleaned from the literature referred to in chapter 3 led to a three-step approach to system design.

Firstly, the constituent agents of the system were identified based on consideration of the goals and tasks involved in the compressor application, the related information requirements, and the physical constraints of the machine. The objective of this first step was to identify agents, which are able to achieve goals independently. Some goals require the action of more than one agent therefore the second step was to consider the intended control strategies and thereby clarify the interaction required between agents to achieve the respective goals. Specifically this meant deciding methods of co-operation and the supporting modes of communication. The first two steps of the design process effectively determined the architecture of the MAS. In the third step, the internal design of agents was addressed in terms of structure and modules.

In practice, the above procedure is iterative and a number of potential solutions may emerge depending on the trade-offs made between each of the steps involved. However, for ease and clarity of reporting the three step framework is retained.

7.4.1 Constituent Agents

To begin with, the principal goals and related control tasks for the IGC application were reviewed. The control tasks were categorised in terms of perception, cognition and execution, as previously defined, and a relative priority level was assigned to each of the main goals. The outcome of this review is summarised in Fig 7.1, which constitutes an initial requirements specification for the agents.

A minimum set of agents were identified immediately based on a physical decomposition of the system (as advocated by Parunak et al 1999) and, in so doing, satisfied the physical constraint that each stator row must be capable of independent adjustment. Thus a set of 6 'row agents' were introduced, one for each variable stator row, which carry out, at least, the basic task of setting the stator rows to the 'required' stagger angles. The need for other agents then depends on the extent to which the row agents are able to support the control tasks involved in achieving the system performance objectives. This was decided by the following task analysis.

a) Stator stall margin control

This objective is local to each stator row and requires adjustment of stagger in response to changes in the local flow conditions at the stator arising from changes in the setting

Fig 7.1 IGC Goals and Related Control Tasks

Goal	Sub-Goal	Control Tasks (by type)		
		Perception	Cognition	Execution
Achieve positive values of overal stall margin under all operating conditions. **Priority 1** (Highest)	Achieve positive value of stall margin for individual stator	Determine current value of stator stall margin Determine expected future value of stator stall margin	Determine adjustment of stator to avoid negative stall margin Determine adjustment of stator to recover positive stall margin	Implement required change in stator angle
	Achieve positive value of stall margin of all rotors	Determine current values of rotor stall margins Determine expected future values of rotor stall margins	Identify which stators to adjust Determine adjustment of selected stators to avoid negative stall margins Determine adjustment of selected stators to recover positive stall margins	Implement required change in selected stator angles
Achieve set-point for delivery pressure. **Priority 2**	Achieve complementary changes in outlet pressures of each stage	Determine set-point pressure error	Identify which stators to adjust Determine adjustment of selected stators to eliminate pressure error	Implement required change in selected stator angles
Achieve maximum overall efficiency. **Priority 3**	Achieve complementary changes in efficiency of each stage	Determine value of efficiency and sense and magnitude of change	Identify which stators to adjust Determine adjustment of selected stators to increase efficiency	Implement required change in selected stator angles

of the throttle or of other variable stator rows. Stall margin control has the highest priority and the requirements indicated in Fig 7.1 include stall 'avoidance' and 'correction'. In the latter case response time is critical to minimise the extent and duration of excursion into stall conditions. For these reasons, the control action should be as direct as possible. Because both the control variable (stator stall margin) and the manipulated variable (stator row setting) are uniquely available then it is possible for row agents to execute this task autonomously. This would give the best performance and therefore the task of stator stall margin control was assigned to row agents. The control approach can be the same as that adopted for the single agent system demonstrated in Phase1.

b) Rotor stall margin control

The requirements for this objective are basically the same as for stator stall margin control and therefore it is again desirable for control action to be as direct as possible. The flow analysis in the Appendix indicates that the stall margin for a rotor row is determined by the setting of the stator row immediately upstream and the local flow velocity. Since the latter depends on the settings of all stator rows and the throttle then the relationship may be expressed as:

stall margin of rotor row k+1, $smr_{k+1} = f\{\beta_k, [\beta_i]_{i \neq k}, G\}$

where $[\ \beta_i\ ]_{i \neq k}$ = set of stator angles excluding k

and G = throttle setting

Therefore, given the availability of rotor stall margin information, the necessary control action can be performed, autonomously, by the row agent associated with the upstream stator row. However, should such action be in conflict with that for stator stall margin control then action by other row agents will be required to bring about a sufficient change in the local flow velocity.

In assigning this task to row agents it was also assumed that the cognitive effort involved would not over-complicate the internal design of the agents or prejudice their other tasks. If the control rules are similar to those for stator stall margin control then this should not be the case. Otherwise the introduction of separate agent(s) would need to be considered.

c) Delivery pressure regulation

This is a system-level objective which involves the collective action of all row agents in response to changes in delivery pressure set-point or throttle setting in order to achieve, or maintain, the required delivery pressure. It is desirable for control action to be fast in response to throttle changes but this is probably not important when dealing with set-point changes. It was decided that during the control action it is not necessary to co-ordinate the relative adjustment of stator rows explicitly. (However, the independent action of row agents in avoiding local stall conditions was expected to regulate relative adjustments to some degree.) Instead, the strategy follows that of Reiss and Blöcker referred to in chapter 2, in that changes to operating point are made as quickly as possible and then, once at the new operating point, an optimisation process is applied.

Since delivery pressure is not a local control variable for row agents it may seem appropriate to introduce a separate agent to perform the task. However, without the need to co-ordinate stator adjustments the task is simplified and can be carried out independently by row agents. Whilst this has the disadvantage of having to make available to all row agents the set point and delivery pressure information it has the advantage that conflicts arising with other system objectives can be resolved 'within' individual row agents rather than through inter-agent communication. The proportional control approach used in Phase 1 was again used for this task.

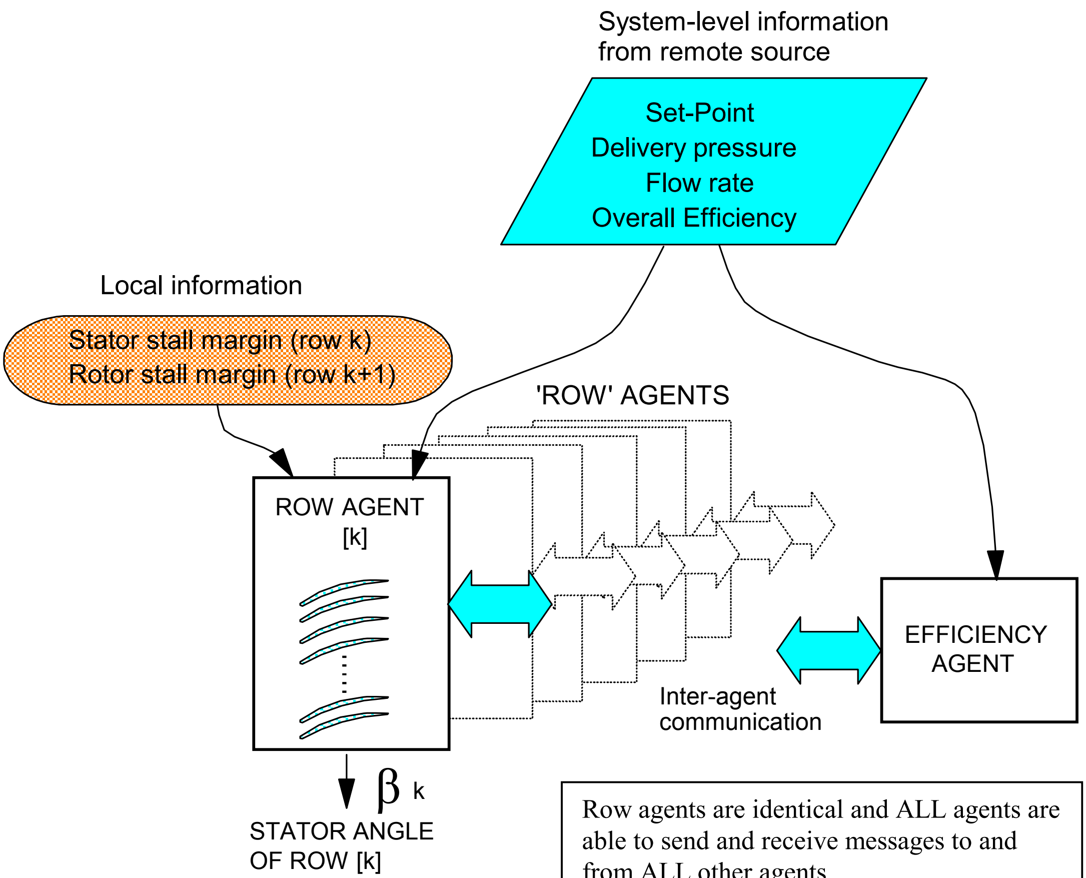
d) Optimisation

This is another system-level goal and specifically refers to maximising overall efficiency at a particular operating point when steady-state conditions prevail. Due to the independent action of row agents it is expected that stator row settings would not necessarily be optimal following a change in operating point. (The existence of an optimal configuration of stators for a given operating point was a conclusion of Phase 1.) The optimisation task therefore requires the determination and application of optimal stator row settings.

The method proposed for optimisation is based on that investigated in Phase 1 which involves the incremental adjustment of each stator in turn whilst monitoring the effect on overall efficiency. It should be noted that, by definition, optimisation is not a time critical task and also does not compete with other tasks. Since a major feature of the optimisation process is the repeated selection of row agents for co-operation then the introduction of a 'specialist' agent seemed appropriate. This would also facilitate the development and trial of more sophisticated optimisation routines as may be required, depending on the results of simulation trials. Therefore, it was decided to include an 'efficiency' agent in the MAS design.

The constituent agents and related information requirements are shown in the diagram below.

Fig 7.2 Constituent Agents



7.4.2 Agent Interaction

The nature of the interaction between agents depends on control strategies proposed for achieving the system objectives. From the task analysis of the previous section, the only control strategies, which involve explicit interaction between agents, are those for rotor stall margin control and efficiency optimisation. The interaction requirements for these are considered further here. The method of communication chosen to support interaction between agents was *message passing* for reasons previously discussed in Chapter 3.

a) Rotor Stall Margin Control

The need for agent interaction arises if a row agent detects a conflict between the corrective action required for local stator stall control and that required for downstream rotor stall control. In such a case the row agent sends messages to all other row agents effectively requesting that they adjust their respective stator row settings so as to increase the stall margin of the particular rotor row in question. The row agent repeats the transmission of messages until the rotor row stall margin is once again of positive value. There is no need for the receiving agents to send any return message.

It is possible that conditions might arise in which more than one row agent detects such local conflict and thus other row agents will receive messages requesting help from more than one source. However, the action required of the receiving agent is the same regardless of the source of the request so this is of no consequence.

Since the content of the message sent by a row agent in the above situation is the same for all destinations then the appropriate mode of communication is *broadcast.*

b) Optimisation

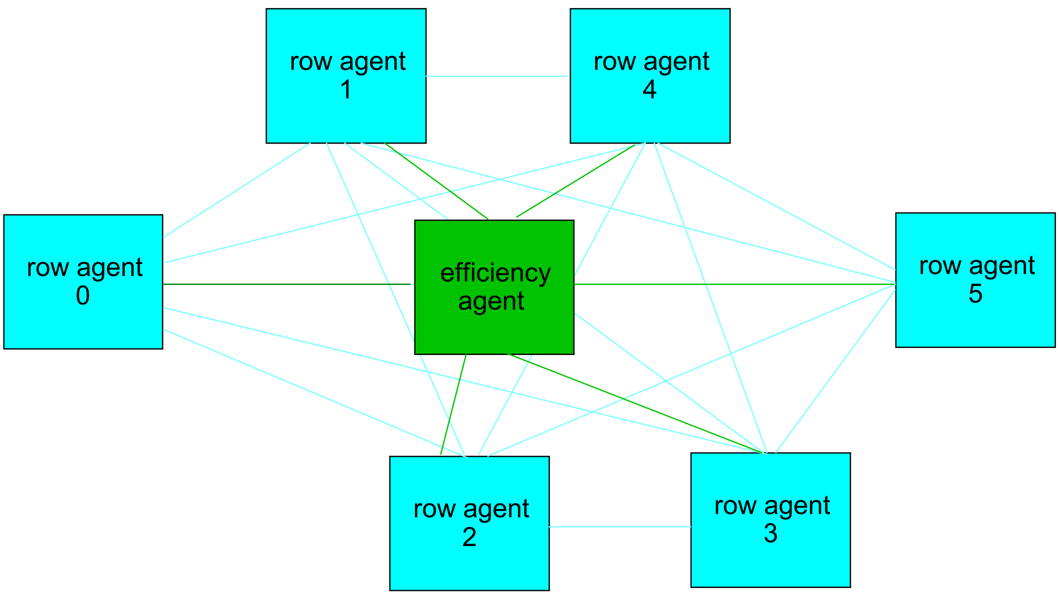
Optimisation requires interaction between the efficiency agent and the row agents. The process commences when a stable operating point has been reached, or more significantly, when the row agents are in a steady-state condition. Therefore it was proposed that when a row agent no longer detects conditions requiring corrective action it sends a message to the efficiency agent indicating its internal state. When the efficiency agent has received such a message from all row agents then it will initiate the optimisation process.

Although the method for maximising efficiency was not known in detail at this point, it was evident that it would require messages to be sent from the efficiency agent to particular row agents requesting action to be taken. Also, it would probably be necessary for the efficiency agent to send a message to all row agents to signal completion of the optimisation process. So the modes of communication required to support optimisation are *direct point-to-point* and *broadcast*.

7.4.3 MAS Architecture

Based on the preceding sections, the MAS architecture proposed for the IGC application was conceived as an *agent network* comprising 6, similar, row agents and a single efficiency agent. The simplified system diagram below shows the agents and their communication links.

Fig 7.3 MAS Architecture



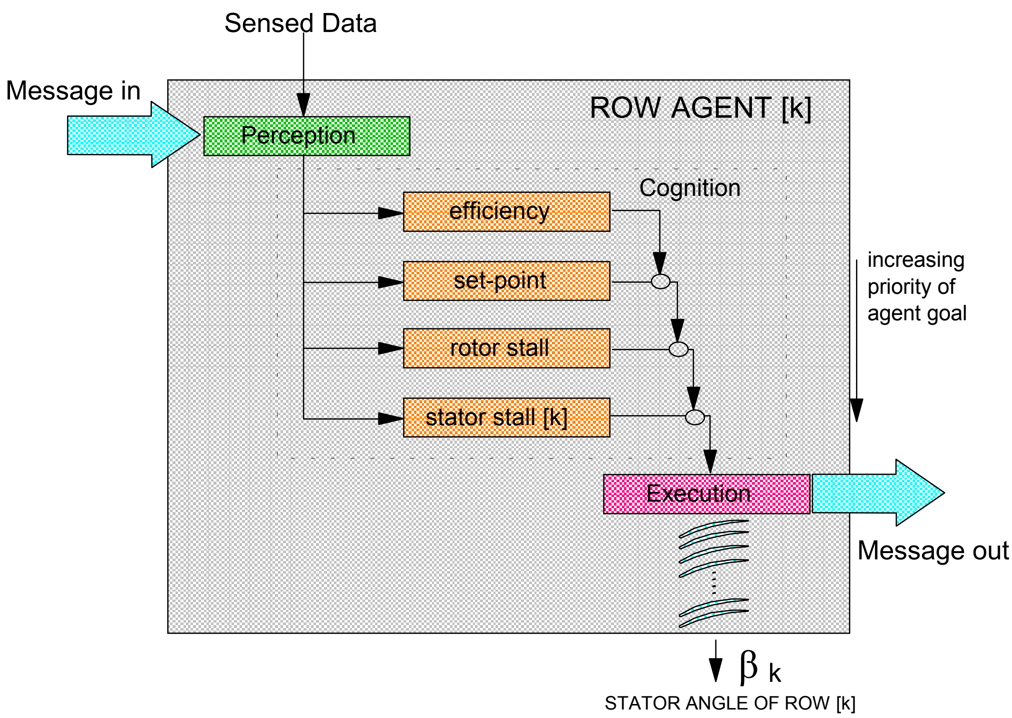
7.4.4 Agent Design

From an agent classification point of view the constituent agents of the above MAS are of the simplest type, being described by Wooldridge (1999) as 'situated reactive' in that they are embedded in the system environment and react to sensed changes in that environment. The basic internal structure of agents follows the simple *perception, cognition, execution* model used previously. The main internal modules of the two types of agent, i.e. row and efficiency, are identified in the following sections.

a) Row Agent

The row agent is required to support several goals. This coupled with the decision to adopt a pre-determined order of priority for the goals, leads naturally to a *subsumption* architecture of the type proposed by Brooks (1986). This is shown in the diagram below.

Fig 7.4 Row Agent Architecture

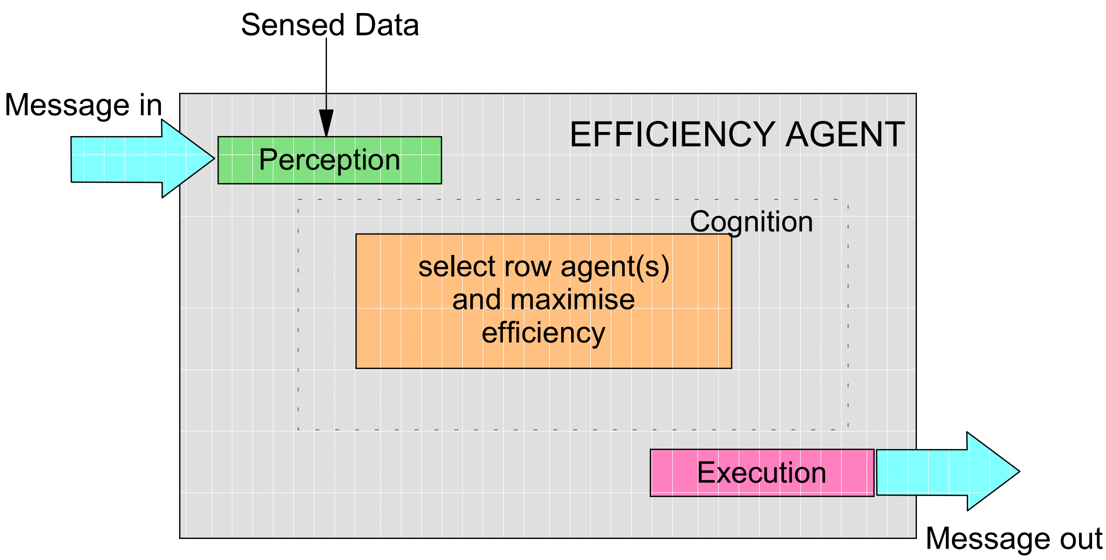


In this design the cognition module comprises a set of task modules each of which is invoked according to the assigned priority. Therefore, the local goals of maintaining a positive value of stall margin for both stator and downstream rotor are actioned first.

b) Efficiency Agent

The proposed architecture of the efficiency agent is shown below.

Fig 7.5 Efficiency Agent Architecture



The cognition module of the efficiency agent is concerned only with the selection of row agents to which it sends messages requesting action to support the optimisation process.

7.5 Simulation Program

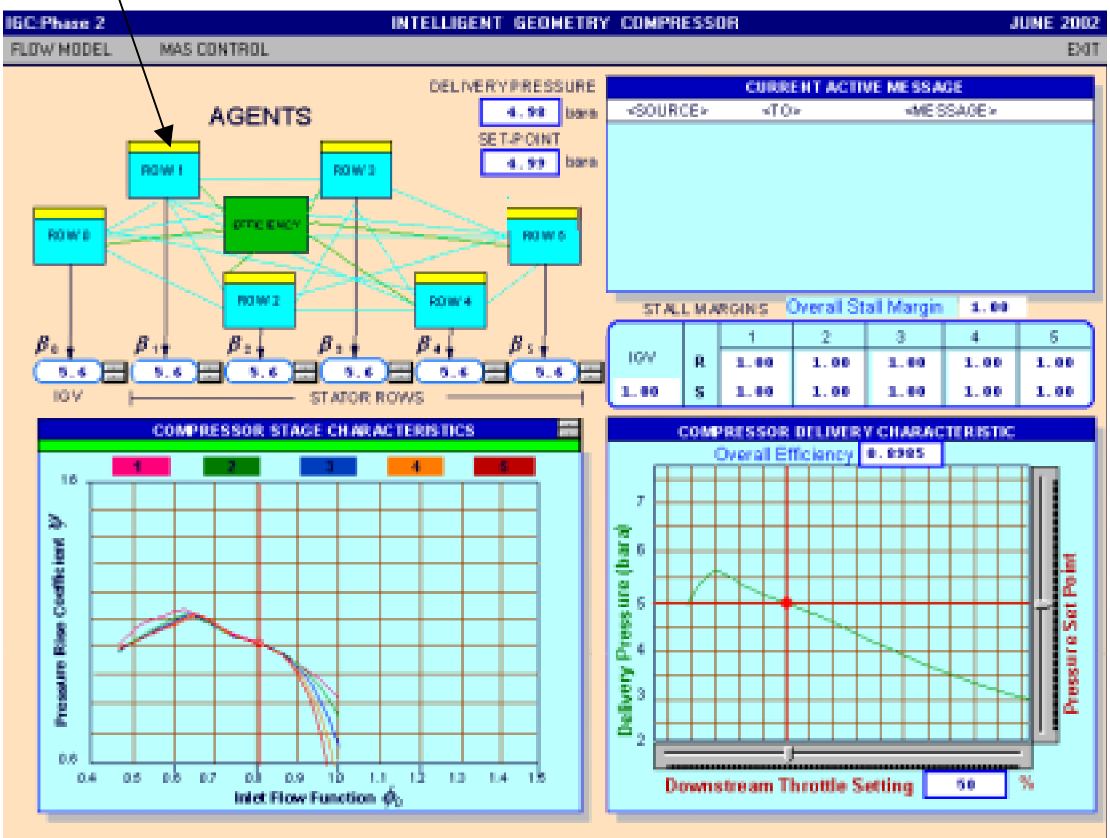
The simulation program, IGC1, developed in Phase 1 was used as the basis for a new program, IGC2, in which the MAS design described in the previous section could be implemented and evaluated. Whilst retaining the original program structure and much of the source code, IGC2 includes additional features necessary to support the simulation of the multi-agent system. These are described below.

7.5.1. Main Program and User Interface (Module:*IGC2main*)

The main change here is in the user interface. Since the need to vary rotational speed is not required in this phase of the work then the user speed control feature is omitted and replaced by a message table, which displays current message passing activity between agents. The schematic of the MAS is also changed to reflect the constituent agents included in the simulation. Otherwise the user interface is the same as that for IGC1. A screen image of the simulation window is shown below.

colour code reflects
agent internal status

Fig 7.6 Phase 2 Simulation Window



7.5.2 Flow Model (Module: *Runflow*)

The flow computation was unchanged and only minor changes to ancillary routines were made to support multi-agent operation. These include additional 'sensor update' event flags necessary to extend program thread synchronisation to all agent threads. Also, additional global variables were introduced to enable agents to access the required control variables generated by the flow model. Finally, code was added to the 'results' module to update the message table referred to above.

7.5.3 MAS implementation

a) Constituent Agents

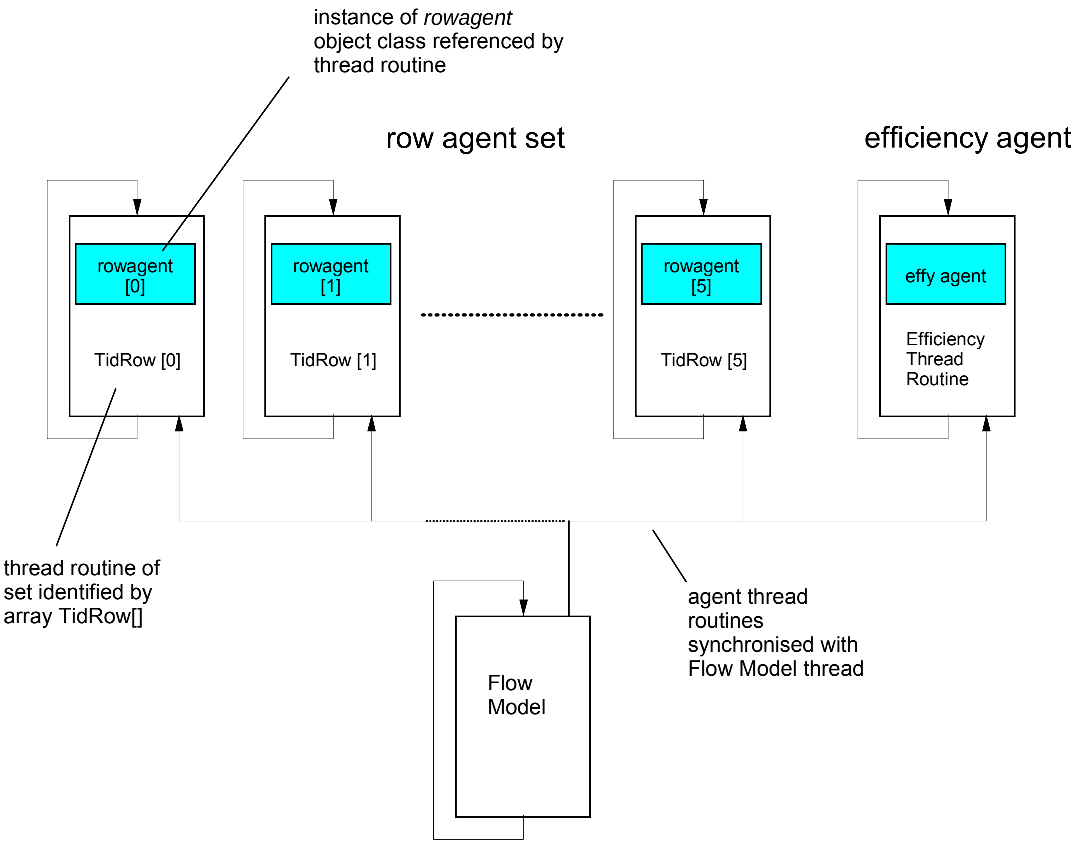
As previously described, an agent is implemented in the program code as a C++ object whose methods are called by instructions in a uniquely assigned program thread. The proposed MAS design requires a set of row agents and a separate agent for optimising overall efficiency.

Because the operation of row agents is virtually identical the creation of the necessary program threads and objects is much simplified. The objects associated with row agents are declared as a 6-element, 1-dimensional array of the object class *rowagent.* Similarly, the program thread identifiers are also declared as an array and this enables program threads to be created by means of a simple loop routine. On creation of each thread, a common thread routine is launched whose argument is the array index of the related thread identifier. This index is used in the thread routine to identify the particular instance of *rowagent* to be referenced when calling object methods. In this way, 6 independent row agents can be activated which run concurrently with one another and the flow model.

The efficiency agent has its own object class definition and corresponding program thread and thread routine.

The overall implementation of the MAS may be visualised as shown in Fig 7.7 in which agents are depicted as instances of object classes referenced within separate concurrent program threads. The whole set of agent thread routines is synchronised with the flow model routine for the reasons, and in the way, previously described in Chapter 5.

Fig 7.7 Overall Implementation of MAS



To aid the visualisation of the MAS operation, a colour code is used on the agent graphic to signify the internal state of the agent. The code is defined as follows:

AMBER: steady state, no control action

GREY: stator stall margin corrective action

BLUE: downstream rotor stall margin corrective action

RED: overall rotor stall margin corrective action

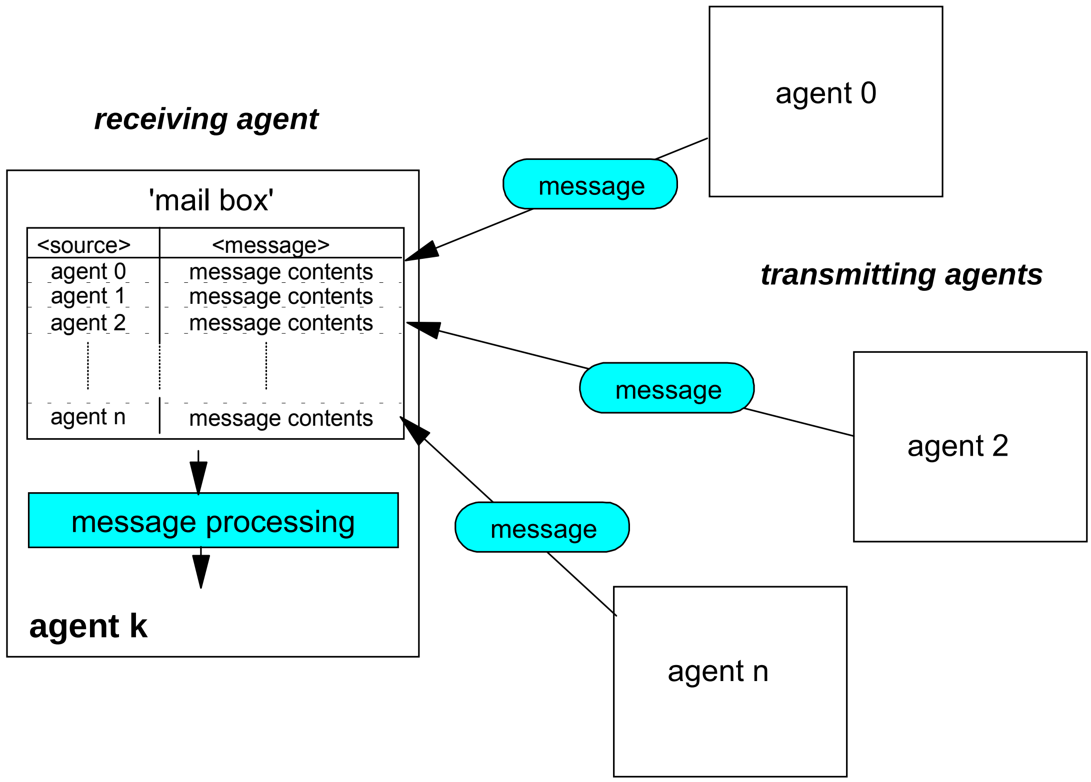
GREEN: set-point error corrective action

PURPLE: optimisation of efficiency

b) Message Passing

The MAS design requires explicit interaction between agents and therefore an appropriate scheme for message passing needs to be introduced. Conceptually, in the scheme adopted here each agent has a 'mail box' into which other agents pass messages as required. An agent checks its mail box on every cycle of operation and processes the contents. To cater for the eventuality of receiving messages from more than one agent during the period of an operation cycle the mail box is divided into sections, one for each of the agents in the system. All messages in the mail box are read and processed according to the message contents. The scheme is illustrated in Fig 7.8.

Fig 7.8 Concept of Message Passing Scheme



To implement the message passing scheme required the definition of a suitable message format and methods for message transmission and processing.

The message format adopted here comprises four fields and is defined as follows:

message = {<source> <priority> <message_id> <value>}

where:-

<source> : identifies the transmitting agent (integer)

<priority> : signifies the priority ranking of message (integer)

<message_id> : defines the request being made by the transmitting agent (integer)

<value> : parameter associated with request (float)

The message format enables an agent to pass requests for pre-defined actions by the receiving agent. For example, the efficiency agent can request a row agent to 'maximise efficiency' simply by using the appropriate message_id. The priority field is redundant in situations where the priority ranking is implicit in the message_id or in the source identification.

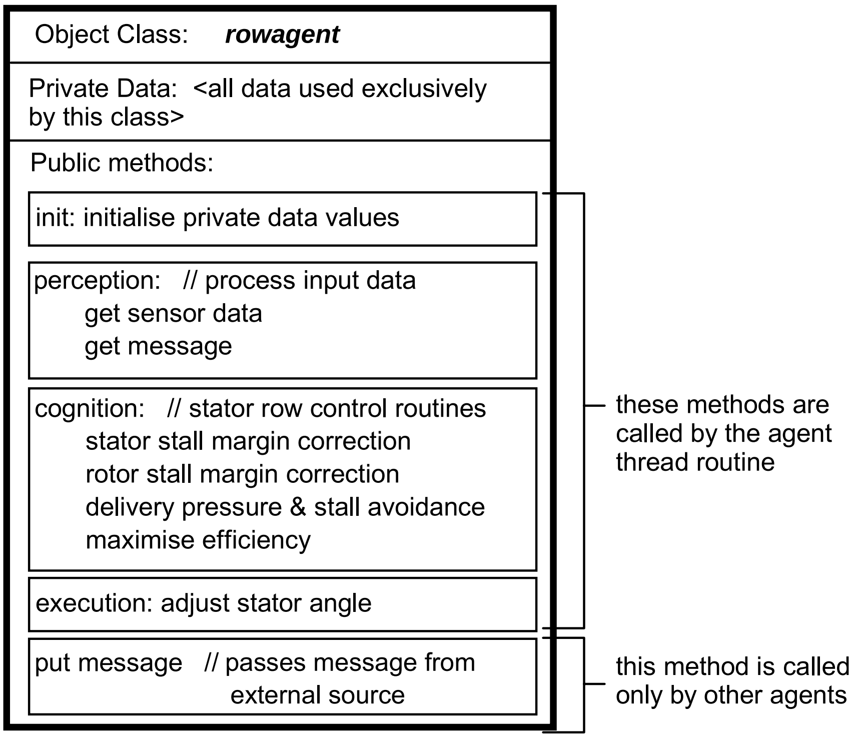
The object method used for transmitting a message is *put_message* and, unlike the other methods associated with an agent, this method is called only by other agents. When an agent decides to pass a message then it calls the *put_message* method of the receiving agent having first encoded the message fields into the corresponding argument parameters of *put_message*. The message fields are then assigned by *put_message* to the elements of a 2-dimensional array in the receiving agent's private data set (effectively the agent's 'mail box') which has been allocated to the transmitting agent.

The second object method introduced is *get_message* and this is called only by the receiving agent. Each agent thread calls *get_message* on every cycle of operation. When called, *get_message* firstly reads the message_id and uses this value to select the appropriate (pre-defined) routine for processing the message. This is repeated for all entries in the message array.

c) Row Agent Methods

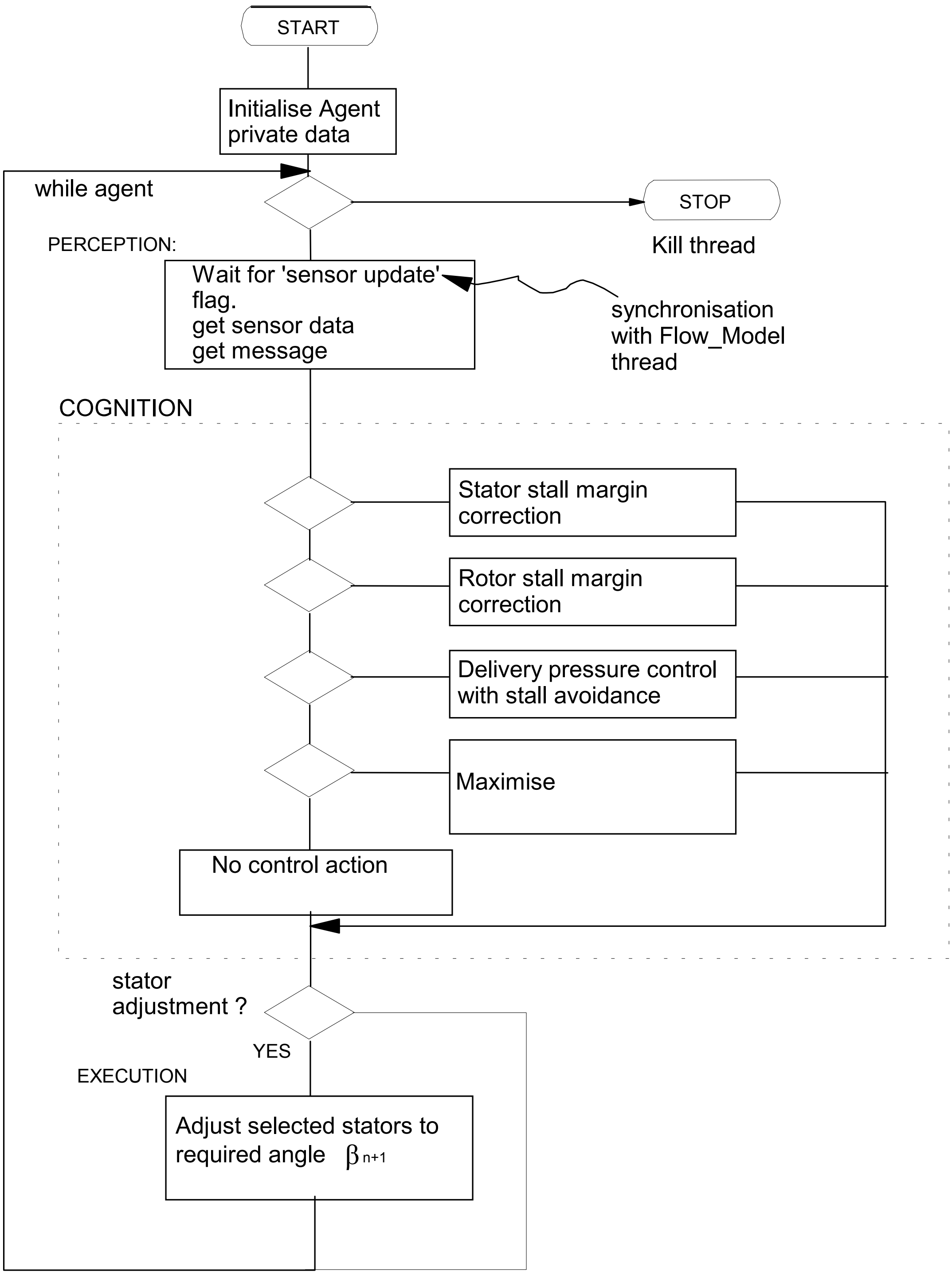
The methods defined in the *rowagent* object class correspond to the internal modules identified in the design of the row agent. The object class definition is shown below and includes methods for initialisation and those for message passing described above.

Fig 7.9 Row Agent Object Class Definition



The architecture of the row agent design is reflected in the control structures of the thread routine and the *cognition* module. These are shown in the flow chart of Fig 7.10. The subsumption-type architecture is implemented by a succession of decision branches (if-else statements) organised in order of priority. The decision criteria, in general, are based on the perceived state of the fluid environment or on requests for action received by messages from other agents.

Fig 7.10 Row Agent Control Structure



The essential part of the row agent behaviour lies in the control routines of the cognition module. The implementation of these is considered in more detail below.

i) Stator Stall Margin Correction

This routine is derived directly from that developed in Phase 1 for global stator control. In this case, of course, the control variable is stator row stall margin rather than overall stall margin. Thus the control rule applied by the j^{th} row agent is:-

$$\beta[j]_{n+1} = \beta[j]_n - \mathrm{sign}(dsms[j]_n) \,.\, \mathrm{sign}(d\beta[j]_n) \,.\, (k.sms[j]_n - \delta)$$

where:

$\beta[j]$ = j^{th} stator row angle

$sms[j]$ = stall margin of j^{th} stator row

d( ) = difference between last and previous values of variable

n+1 - signifies next (planned) value of variable

n - signifies last value of variable

k = gain constant

δ = small constant to ensure control achieves positive value of sm

The rule is applied independently by each row agent whenever the agent perceives that stator stall margin is less than zero.

ii) Rotor Stall Margin Correction

When the row agent detects the stall margin of the downstream rotor row to be negative then a similar control rule to that above for stator stall margin correction is applied. Thus for the j^{th} row agent:

$$\beta[j]_{n+1} = \beta[j]_n - \mathrm{sign}(dsmr[j]_n) \,.\, \mathrm{sign}(d\beta[j]_n) \,.\, (k.smr[j]_n - \delta)$$

where:

$\beta[j]$ = j^{th} stator row angle

$smr[j]$ = stall margin of $j+1^{th}$ rotor row

other symbols are as previously defined

If the application of this rule is in conflict with the corrective action for stator stall margin control then the agent broadcasts a 'request for action' message to other row agents. The detailed rules applied by an agent following receipt of this message were developed later with the aid of simulation trials.

iii) Delivery Pressure Control and Stall Avoidance

Again, the approach adopted for this control routine is derived from that developed in the Phase 1 simulation work. When responding to a set-point error, the basic principle of the control approach is to determine the next change in stall margin and then moderate the amount of stator adjustment accordingly. In this case, however, with stator rows being adjusted independently by row agents it is necessary to take account of changes in both rotor stall margin and stator stall margin. For a given rowagent, the rotor row of interest is that immediately downstream.

Thus the general rule for control of delivery pressure is given by:-

$$\beta[j]_{n+1} = \beta[j]_n - k.(P_{SET} - P_D)_n \ . \ fo$$

where:-

fo = the lesser of stator and rotor coefficients, fs[j] and fr[j+1]

fr[j+1] = coefficient determined by $[j+1]^{th}$ rotor stall margin conditions

fs[j] = coefficient determined by j^{th} stator row stall margin conditions

The detailed definition of these coefficients is dealt with later.

iv) Maximise Efficiency

This routine is invoked when steady-state conditions exist and a request from the efficiency agent has been received. Once started, the row agent will act independently to achieve an increase in overall efficiency. This will continue until either no further increase in efficiency is detected or until interruption by a higher priority task. The approach adopted is to simply adjust the stator angle incrementally in the appropriate direction.

Thus the control rule is of the form:-

$$\beta[j]_{n+1} = \beta[j]_n + \text{dir}\eta.\ \delta$$

where:-

dirη = sign of change required to increase overall efficiency

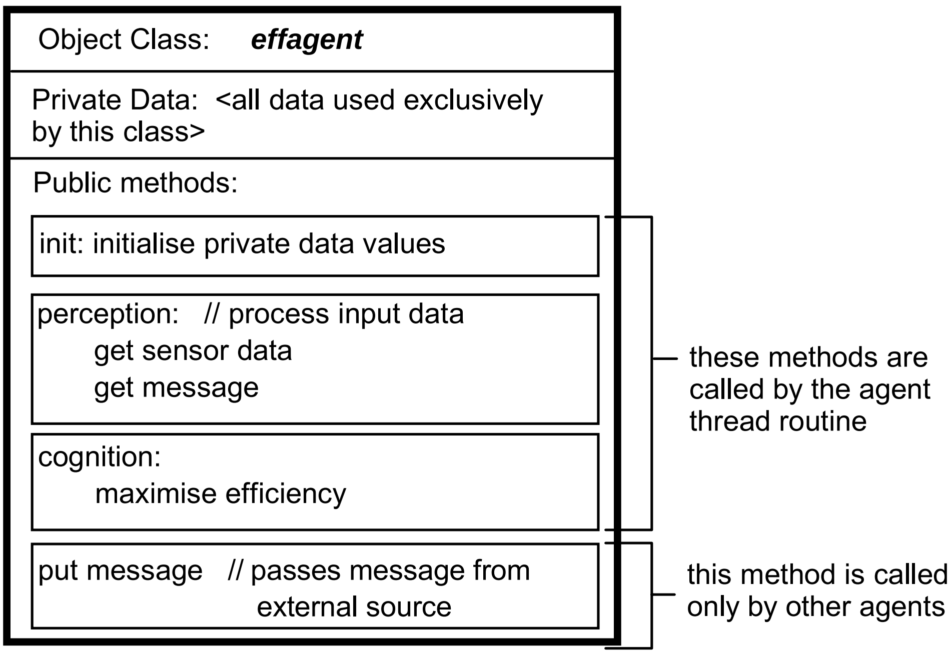
δ = incremental change

On completion of routine, the row agent sends a status message to the efficiency agent.

d) Efficiency Agent Methods

The object class definition for the efficiency agent is given below. It will be noted that there is no separate execution method included since the output from the efficiency agent is a message and is achieved using the *put_message* method of the receiving agent.

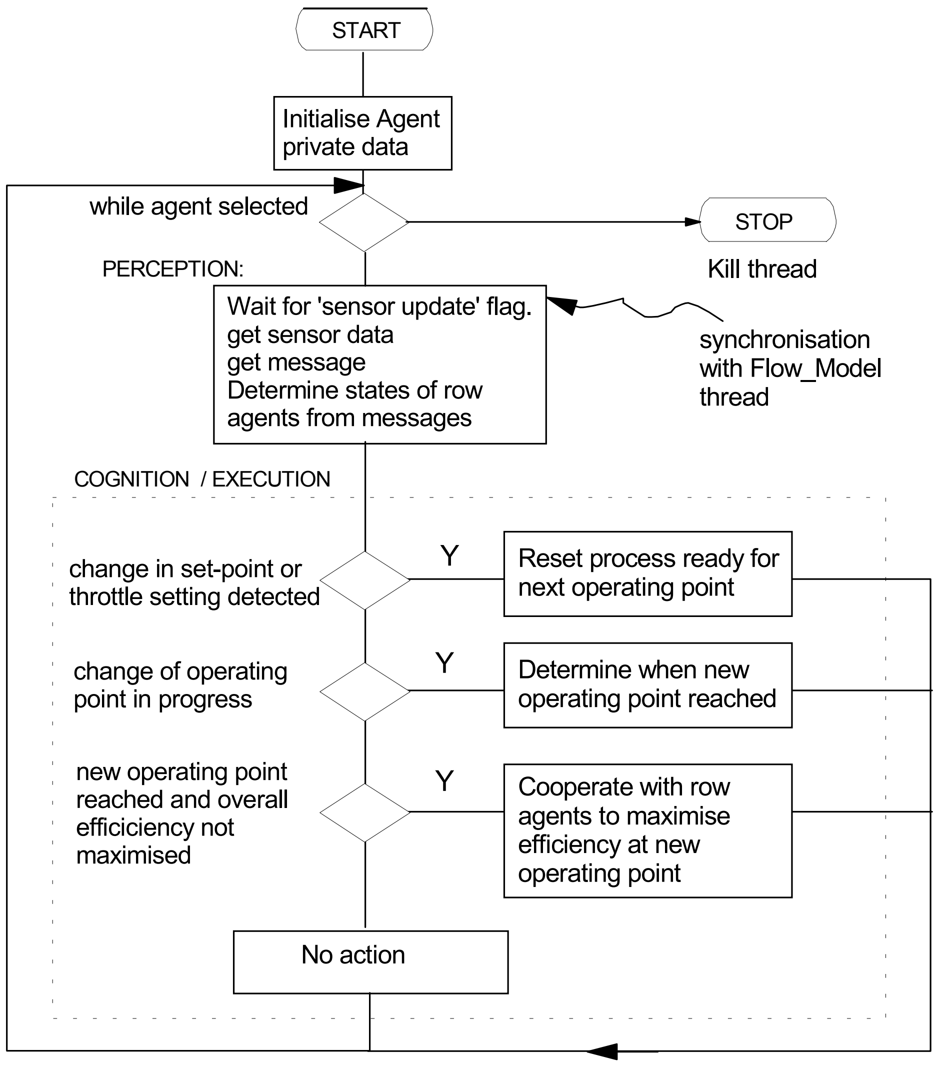
Fig 7.11 Efficiency Agent Object Class Definition



The general form of the control structure of the efficiency agent thread routine is shown overleaf.

The strategy described in the MAS design section was implemented, in which row agents are selected in sequence starting with the agent furthest downstream. The process continues cyclically until no further increase in overall efficiency is detected. During the response of any one row agent, the other row agents continue to act concurrently to maintain the current set point.

Fig 7.12 Efficiency Agent Control Structure



7.6 Simulation Trials

The simulation trials followed a similar pattern to those of Phase 1 for global stator control reported in the previous chapter. Firstly, relationships between the main performance parameters and individual stator row angle were captured by operating the compressor model in 'open loop' mode. The results obtained were used to further develop the control routines initially implemented in IGC2. The second set of trials was carried out using IGC2 in which the compressor model was operated under the control of the multi-agent system.

7.6.1 Parametric Relationships

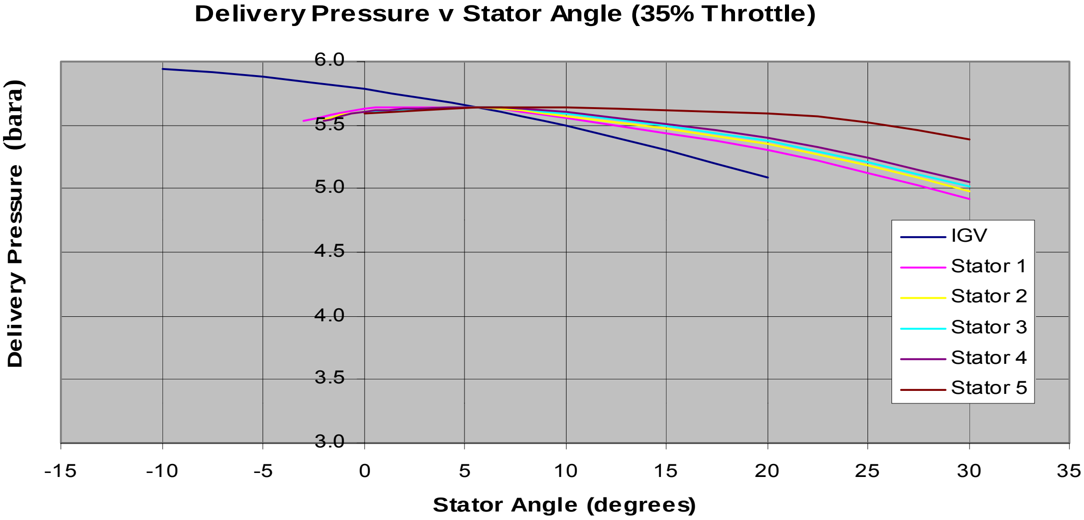
There were three specific objectives for these initial trials. Firstly, it was required to demonstrate the relative contribution of individual stator row adjustment to overall machine performance, specifically delivery pressure and efficiency. The second objective was to establish the potential for improvement in operating range and maximum efficiency resulting from individual stator row adjustment. This would provide a reference in the subsequent evaluation of the MAS. Finally, it was required to establish the relationship between stall margin at given rotor rows and individual stator row angles in order to aid the development of the rules for rotor stall margin control.

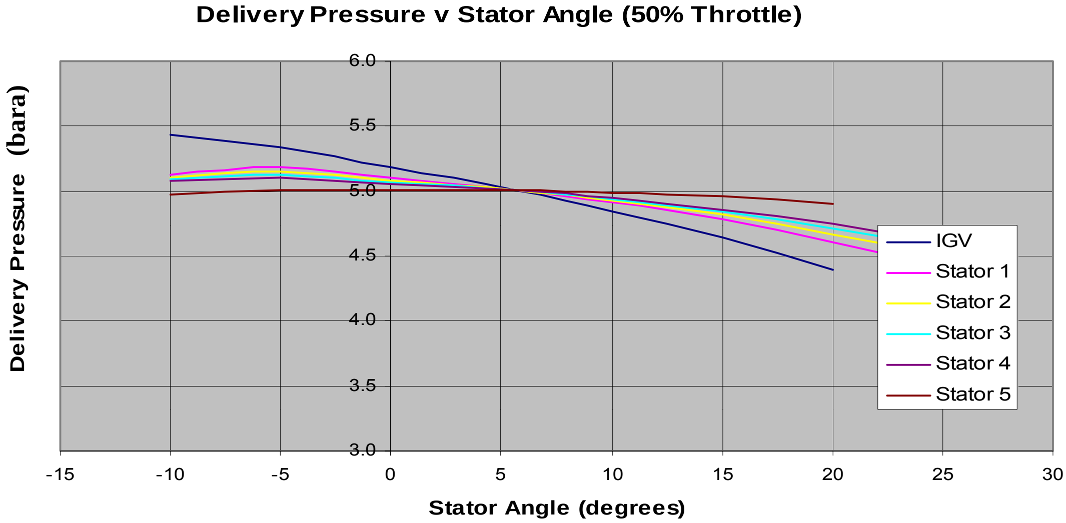
a) Overall Machine Performance

Machine performance depends on the internal flow geometry defined by the settings of the stator rows and throttle. Clearly, many different geometric configurations are possible. To simplify matters, trials were carried out in which each stator row was adjusted over a range of angles whilst keeping the other rows at a constant, reference, angle. These were repeated for a small number of different reference and throttle settings. A selection of results is presented in the following sets of graphs.

Fig 7.13 Delivery Pressure v Stator Angle

Reference angle for other stators = + 5.6 deg (design value)





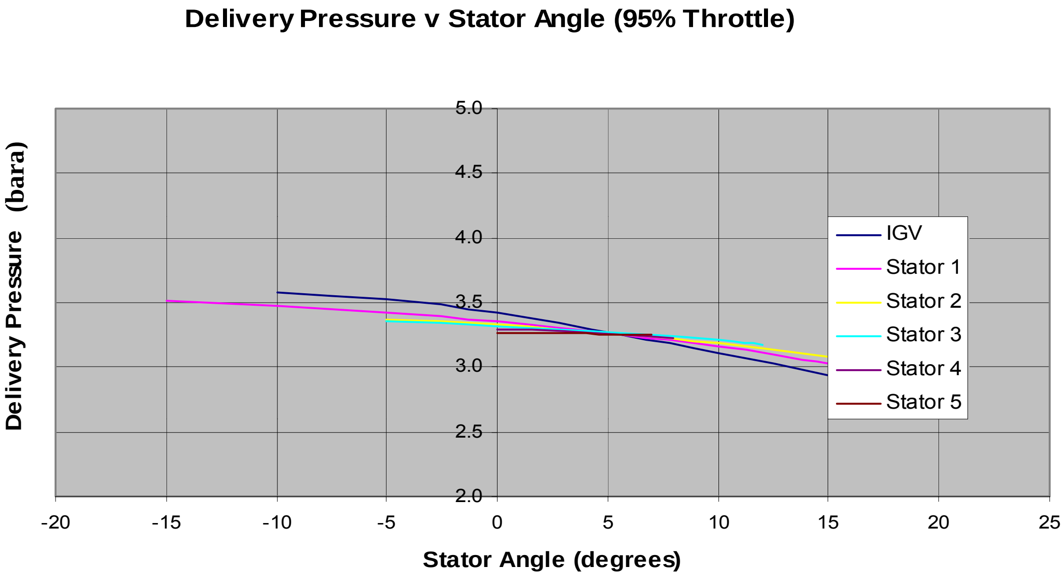


Fig 7.14 Delivery Pressure v Stator Angle

Reference angle for other stators = + 15 degrees

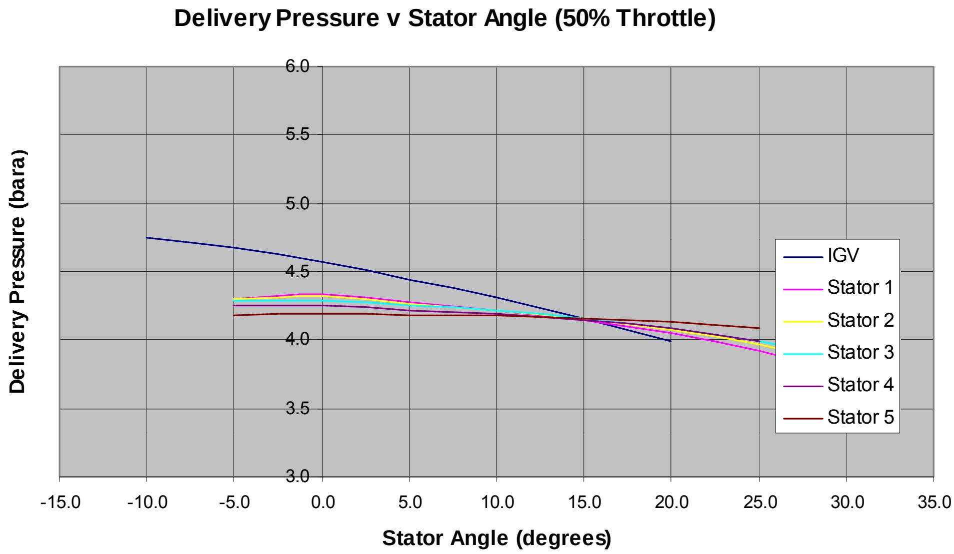


Fig 7.15 Delivery Pressure v Stator Angle

Reference angle for other stators = - 10 degrees

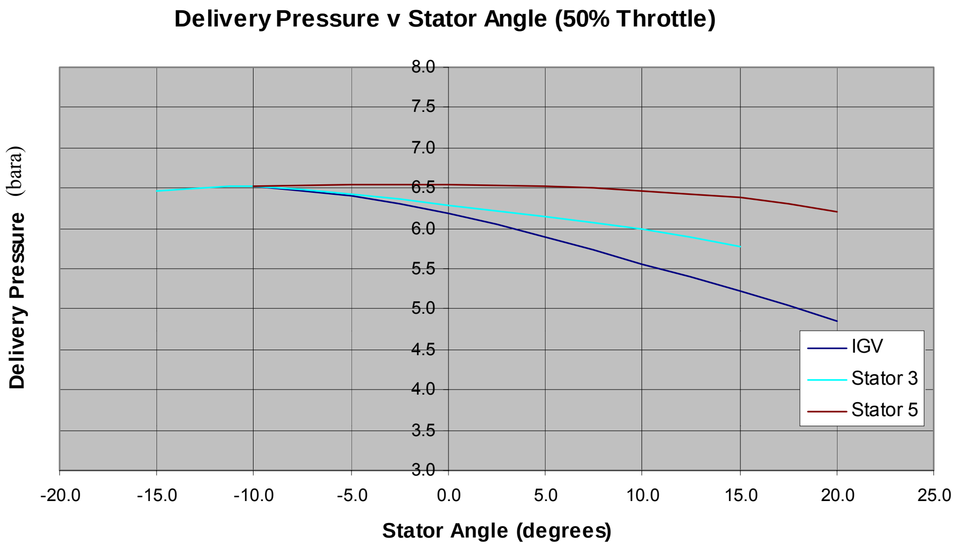
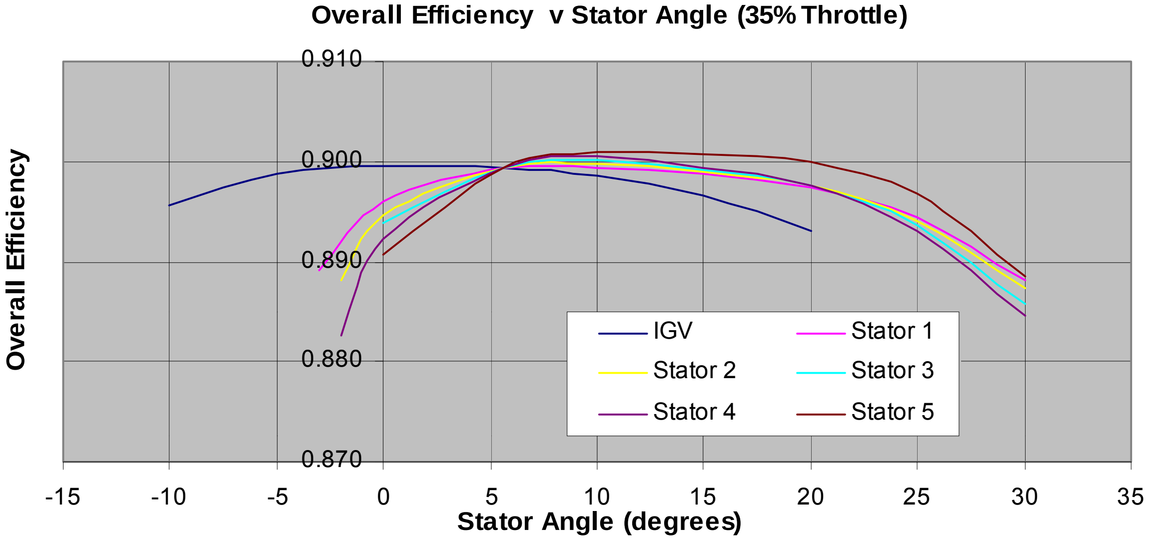
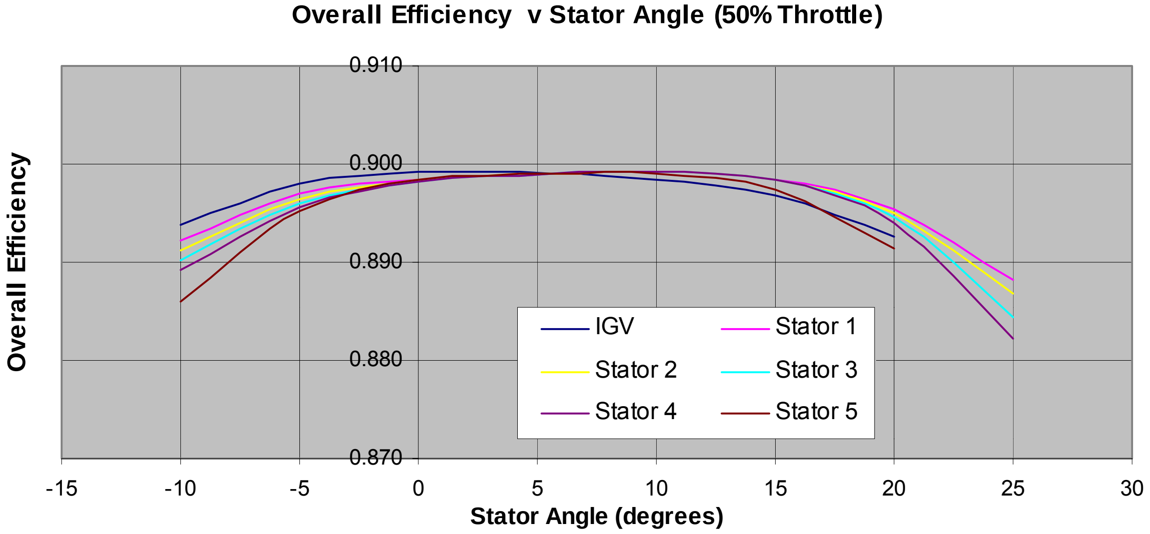


Fig 7.16 Overall Efficiency v Stator Angle

Reference angle for other stators = + 5.6 deg (design value)





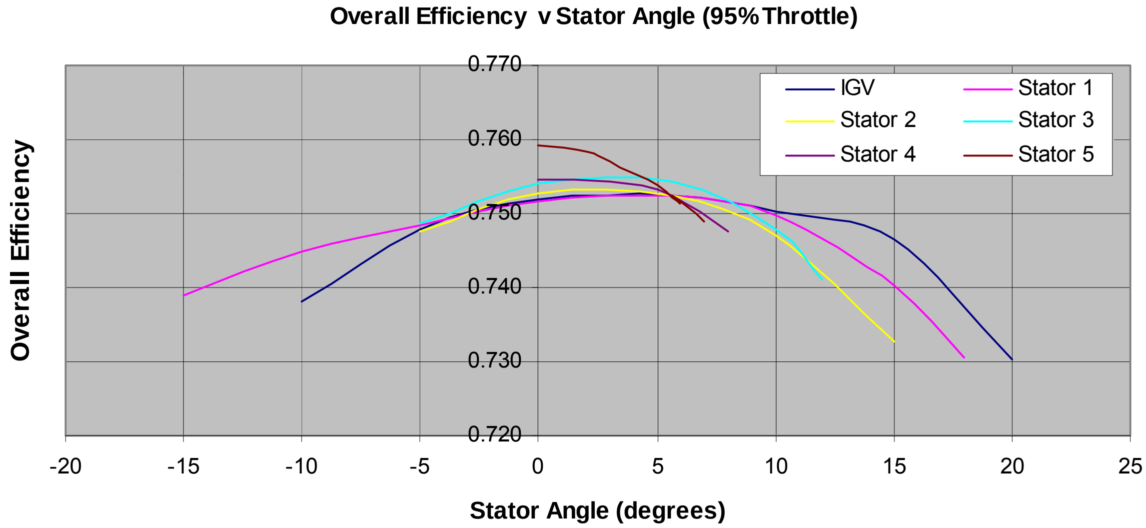


Fig 7.17 Overall Efficiency v Stator Angle

Reference angle for other stators = + 15 degrees

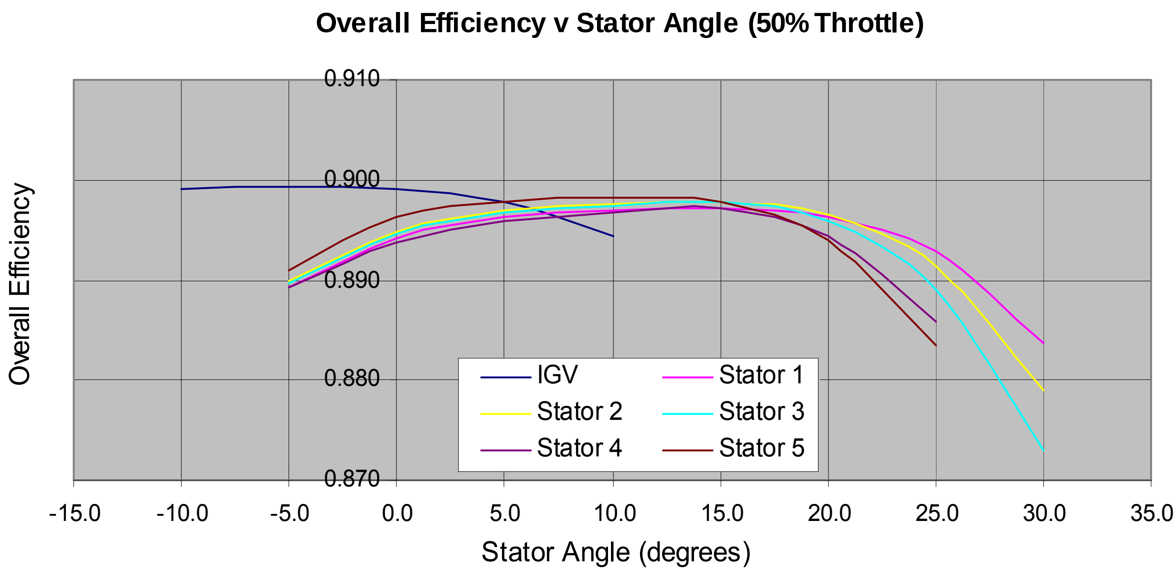
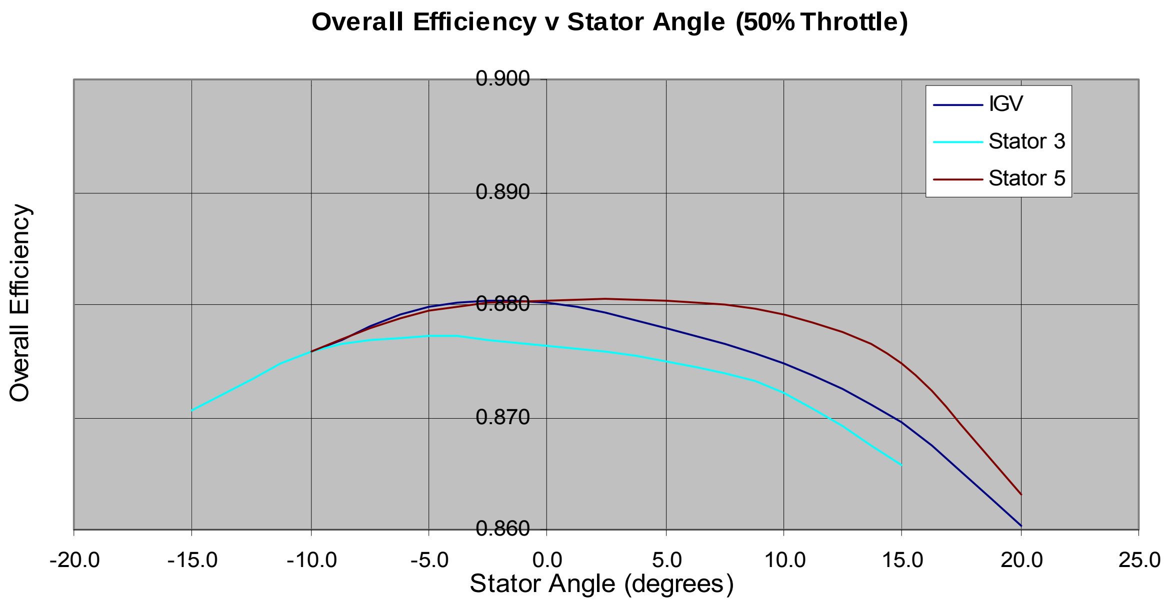


Fig 7.18 Overall Efficiency v Stator Angle

Reference angle for other stators = - 10 degrees



The main observations from the graphs are:-

Delivery Pressure:

- delivery pressure decreases with increasing throttle setting
- general form of characteristics does not vary significantly with throttle setting
- characteristics intersect at stator angle equal to reference angle
- stator rows further downstream have relatively less effect on delivery pressure than those upstream, with the IGV being most effective. This is explained by the multiplicative effect of compressor stages in which a pressure rise at an upstream stage is compounded by each subsequent stage.
- the changes in delivery pressure due to individual stator angle are small compared to the result shown in Phase 1 where all stators were moved together.

Overall Efficiency:

- effect of individual stator variation is small over a large range of angles
- stator rows appear similarly effective over a wide range
- only at points well away from design do changes in efficiency become significant and do individual characteristics start to differ
- the maximum change in overall efficiency due to stator adjustment is about 2 to 3%

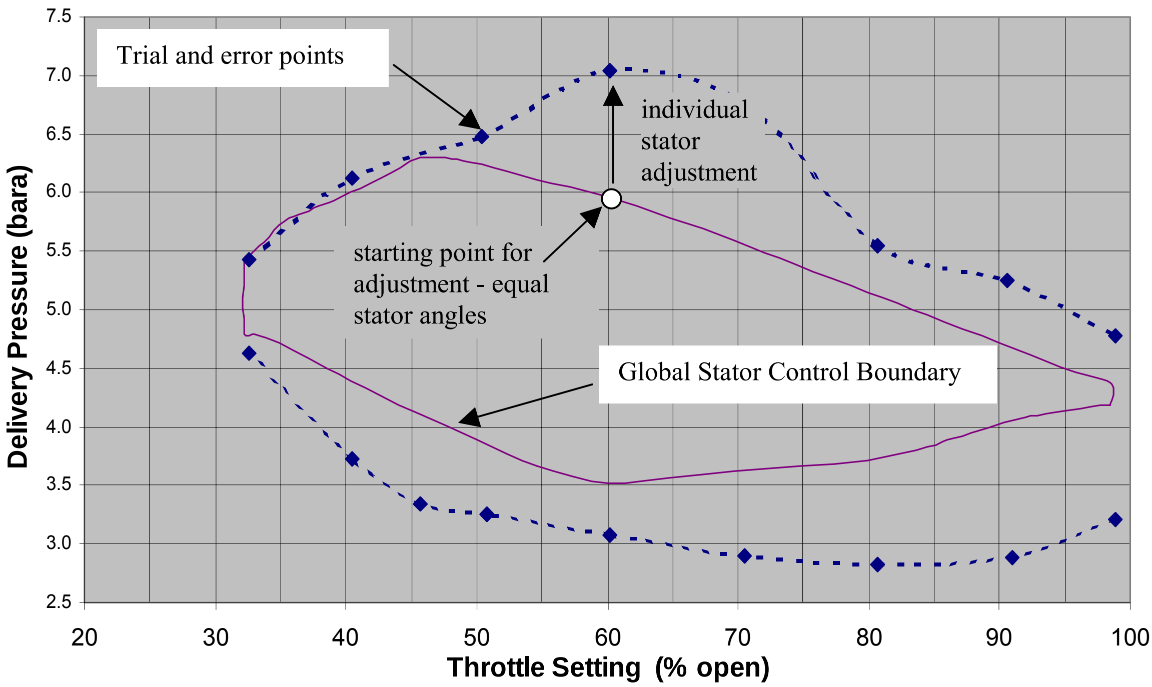
<u>b) Potential Machine Performance</u>

An indication of the performance improvement that may be achieved by controlling individual stator rows was obtained through manual trial and error adjustments. Firstly, the operating range as defined in Phase 1 was considered.

<u>i) Operating range</u>

At selected points on the boundary of the range, stator angles were adjusted in order to achieve the lowest or highest pressure whilst avoiding negative values of rotor or stator stall margin. The results obtained are shown on the plot of Fig 7.19. The general method of adjustment was to first reach a point on the global stator boundary using equal values of stator angle and then to adjust each row in turn by similar amounts (e.g. 2 or 3 'steps' where 1 step = 0.1 degree), repeatedly, starting with the row furthest downstream. The order of adjustment follows the method for setting variable stator rows described by Cumpsty (1989). Generally, it was not difficult to locate points on the extended boundary where stator stall was the limiting condition since the effects on stall margin of a given stator row due to the adjustment of other rows was able to be readily countered by adjustment of the effected stator. Where rotor stall was the limiting condition the adjustment process was more involved. Stall margin of a rotor row was influenced by both up- and downstream stator settings.

Fig 7.19 Potential Operating Range



Although not necessarily the maximum possible, the operating range achieved by manual adjustment of individual stator rows represents a significant increase over that obtained when stators are adjusted equally as in global stator control. The area of the extended range is estimated to be more than 60% greater than the operating range for global stator control.

ii) Overall efficiency

Trials were carried out to confirm the feasibility of increasing overall efficiency whilst maintaining a given operating point. In these trials, stator rows were adjusted in turn starting at stator 5 and the process continued cyclically until no further increase in efficiency occurred. After each small adjustment of a given stator, the IGV was adjusted to compensate for any change in delivery pressure. In this way, three different internal pressure distributions were obtained for the same operating point, each corresponding to a different set of stator angles. For each arrangement the overall stall margin was greater than zero. The operating point was deliberately chosen to be away from design in order to maximise the potential for increase in efficiency. The results are shown in Fig 7.20 and Fig 7.21.

Fig 7.20 Stator Row Settings for Same Operating Point

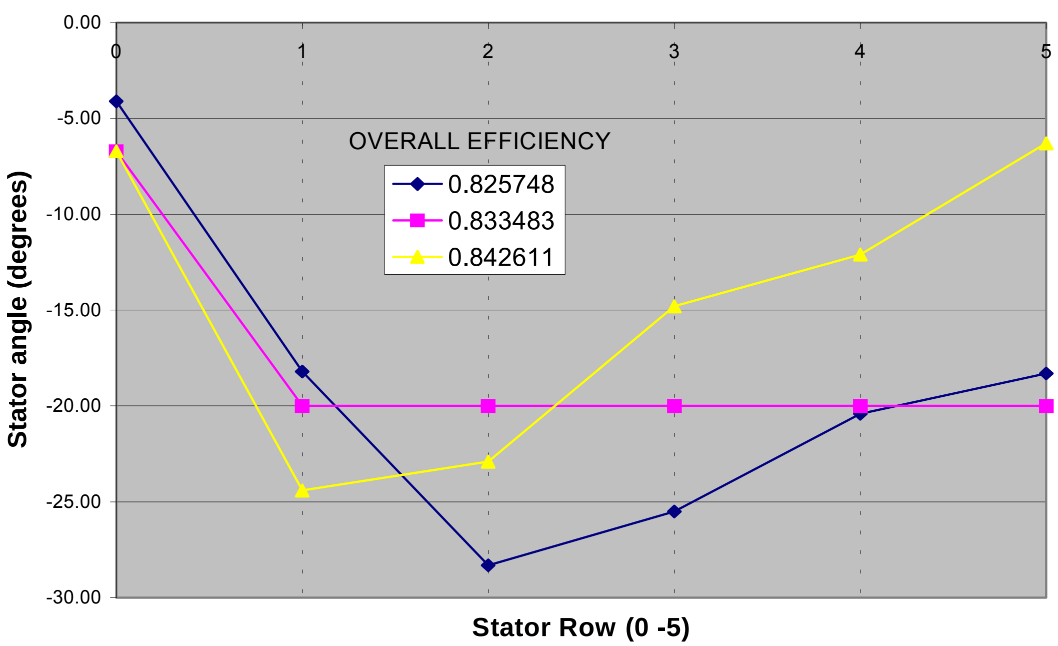
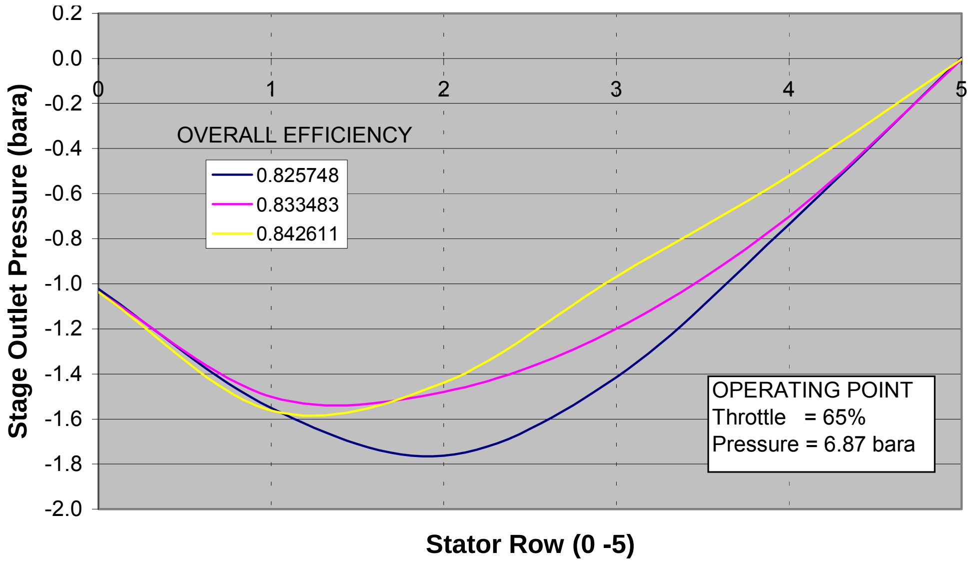


Fig 7.21 Stage Pressure Distributions for Same Operating Point

(Note: Pressures are shown as difference from linear distribution)



The variation in efficiency over the three stator configurations investigated was found to be about 1.7%. This is a significant amount and shows that even if the control objective is constrained to achieve a particular delivery pressure set-point, it is still possible to improve overall performance by "fine tuning" the setting of individual stator rows.

c) Effect of Stator Settings on Stall Margins

As previously explained the stall margin of a particular stator or rotor row represents the degree of incidence of the fluid flow at the inlet of the row. From the analysis given in the Appendix the stall margins for the kth stator and downstream rotor row may be expressed as follows:

stator: $sms_k = f\{\beta_k, \phi_k\}$

rotor: $smr_{k+1} = f\{\beta_k, \phi_{k+1}\}$

where β_k = angle of stator row of stage k

ϕ_k = flow function for stage k

and ϕ_{k+1} = flow function for stage k+1

For a given rotational speed, constant fluid properties and steady flow, the flow function is dependant on the flow geometry determined by the variable stators and throttle. Therefore the above expressions for stall margin may be re-stated as:

stator: $sms_k = f\{\beta_k, [\beta_i]_{i \neq k}, G\}$

rotor: $smr_{k+1} = f\{\beta_k, [\beta_i]_{i \neq k}, G\}$

where [β_i]$_{i\neq k}$ = set of stator angles excluding k

and G = throttle setting

The above expressions indicate the general relationship between stall margins, stator angles and operating conditions. In particular, they show that stator angle is the primary variable for controlling both stator stall margin and the stall margin of the rotor row immediately downstream. However, to support the development of control rules it is necessary to know the sense and relative strength of the relationships. To this end, simulation runs were made for each stator row in which both stator and rotor stall margin values were captured for a range of stator angles. Runs were repeated with different reference values for throttle and non-active stator angles. Thus, strictly, the results obtained represent:

stator: $sms_k = f\{\beta_k\}$

rotor: $smr_{k+1} = f\{\beta_k\}$

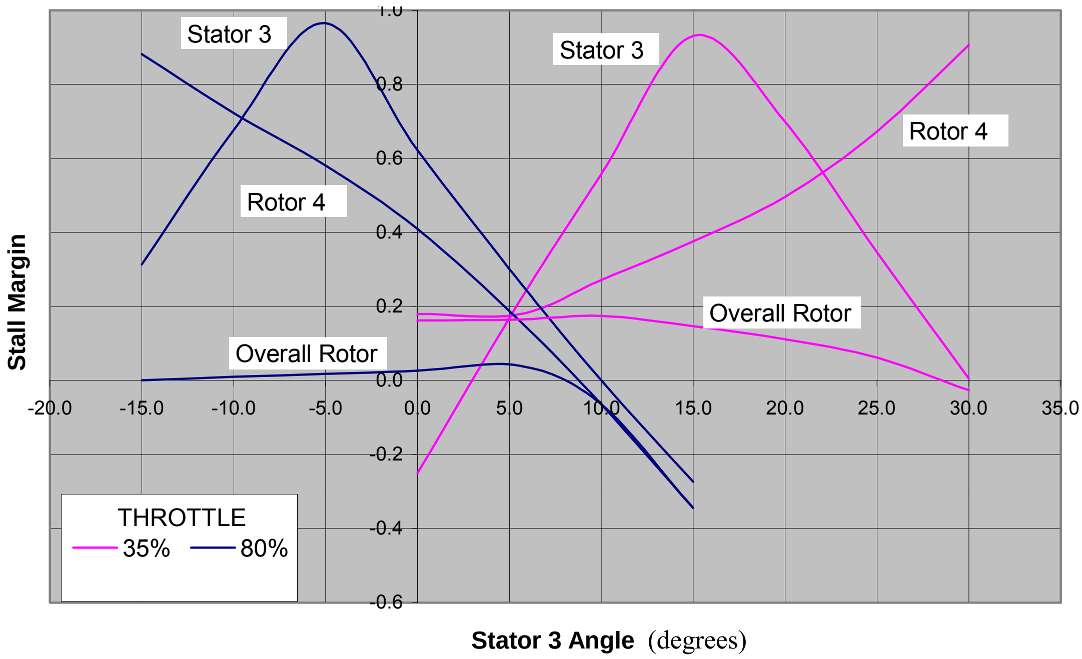
with G and [β_i]$_{i\neq k}$ held constant at selected reference values

In addition, to capture the influence of the adjusted stator on the stall margin of other rotor rows, the variable *overall rotor stall margin* was introduced. This variable is defined as the lesser of the stall margins for all rotor rows.

Firstly, the behaviour of stator 3 is considered, since this is generally representative of an "inner" row i.e. of rows 1 to 4. The graphs of Fig 7.22 show the results for a "low" and a "high" throttle setting whilst other stator rows are held constant at design settings.

Fig 7.22 Effect on Stall Margins of Stator 3 Adjustment

All other stators held at design angle +5.6 degrees



It can be seen that the form of the stator stall margin graph is similar to that obtained in Phase 1 when investigating overall stall margin and global stator angle and has a peak value close to unity. The effect of throttle setting is to shift this graph along the stator angle axis without altering the general form of the graph.

The relationship between stall margin of the downstream rotor and stator angle is clearly a strong one. This also shifts with change in throttle setting so that it has an opposite sense for low and high throttle settings. At low throttle, the rotor stall margin does not approach the zero line anywhere and so rotor 4 is not the limiting row for overall rotor stall. In contrast, at high throttle, rotor 4 stall margin becomes less than zero at stator 3 angles above about 8 degrees. Beyond this point the rotor 4 characteristic coincides with that for overall rotor stall margin indicating that it has become the limiting rotor row (or at least one of them).

The influence of stator 3 on overall rotor stall margin is small, particularly at high throttle setting (until rotor 4 becomes critical) but its effect is not insignificant at low throttle setting.

From the point of view of controlling stator 3 it is important to note that, at least for the conditions considered, there is no conflict between the objectives of maintaining a positive stall margin for stator and for rotor. This is because, when either stator 3 stall margin or overall rotor stall margin become less than zero the sense of the required corrective action is the same for both.

Results for stator 3 at different settings of the other stator rows are given in Figs 7.23 and 7.24.

Fig 7.23 Effect on Stall Margins of Stator 3 Adjustment

All other stators held at +15 degrees

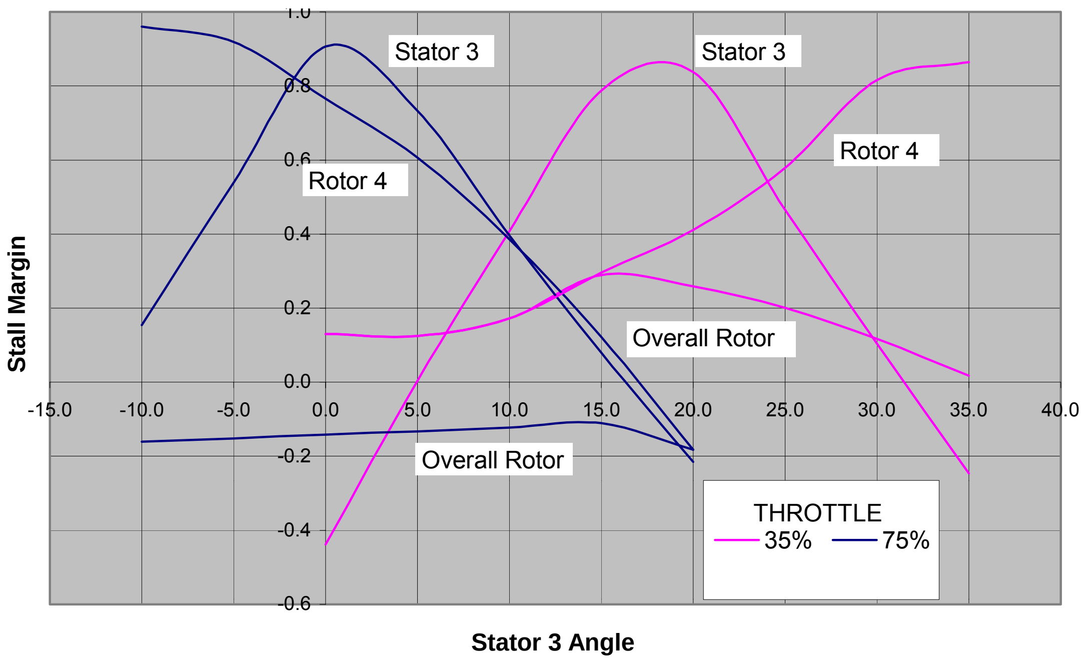
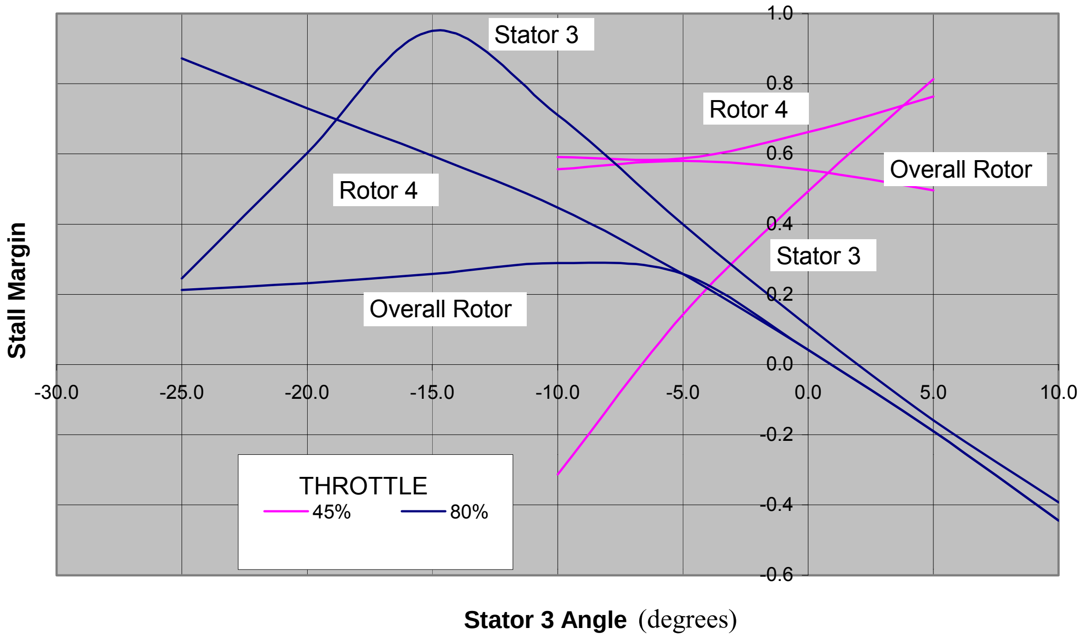


Fig 7.24 Effect on Stall Margins of Stator 3 Adjustment

All other stators held at -10 degrees



The general form of the results is similar to that already described. The principal effect of the different settings for the other stator rows is to displace the various characteristics along the stator angle axis. In addition, the overall rotor and rotor 4 characteristics are shifted in the vertical plane.

The effect of varying the IGV (stator 0) is considered next. The same approach was adopted for capturing and presenting results as was used for stator 3. The results for the three conditions investigated are shown in Figs 7.25, 7.26 and 7.27. The first point to note is that the IGV stall margin characteristic does not shift with throttle setting. This reflects the assumption made when creating the compressor flow model that the flow angle at inlet to the IGV is constant and thus incidence depends only on the stagger angle of the row. This has important consequences when considering the shift in rotor stall margin characteristics that takes place at the extremes of throttle and reference

Fig 7.25 Effect on Stall Margins of Stator 0 (IGV) Adjustment

All other stators held at design angle +5.6 degrees

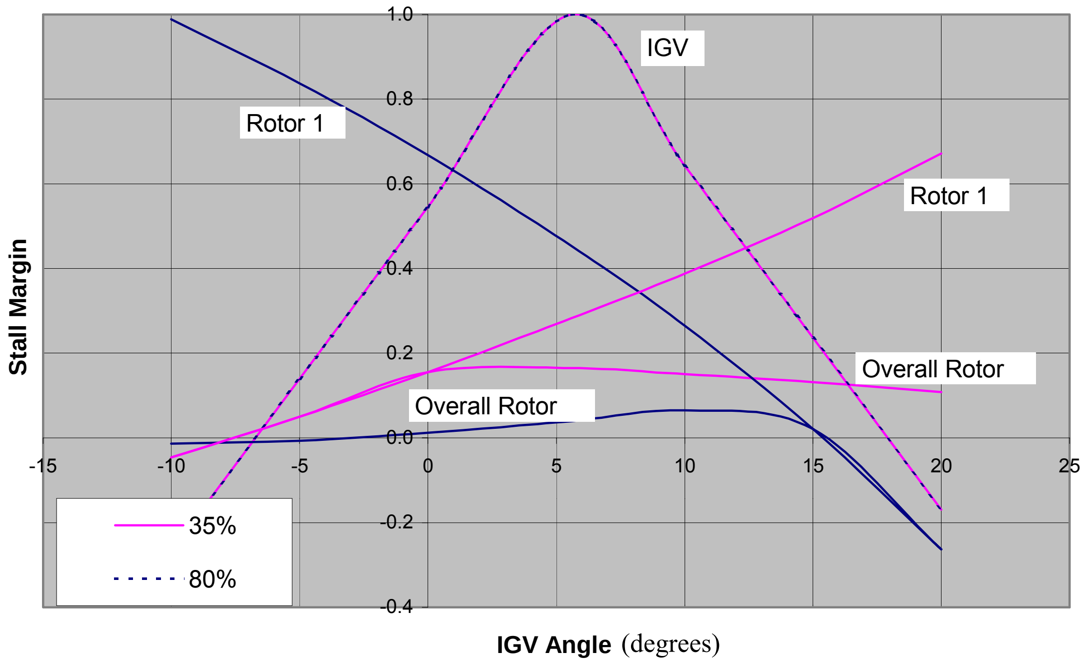


Fig 7.26 Effect on Stall Margins of Stator 0 (IGV) Adjustment

All other stators held at +15 degrees

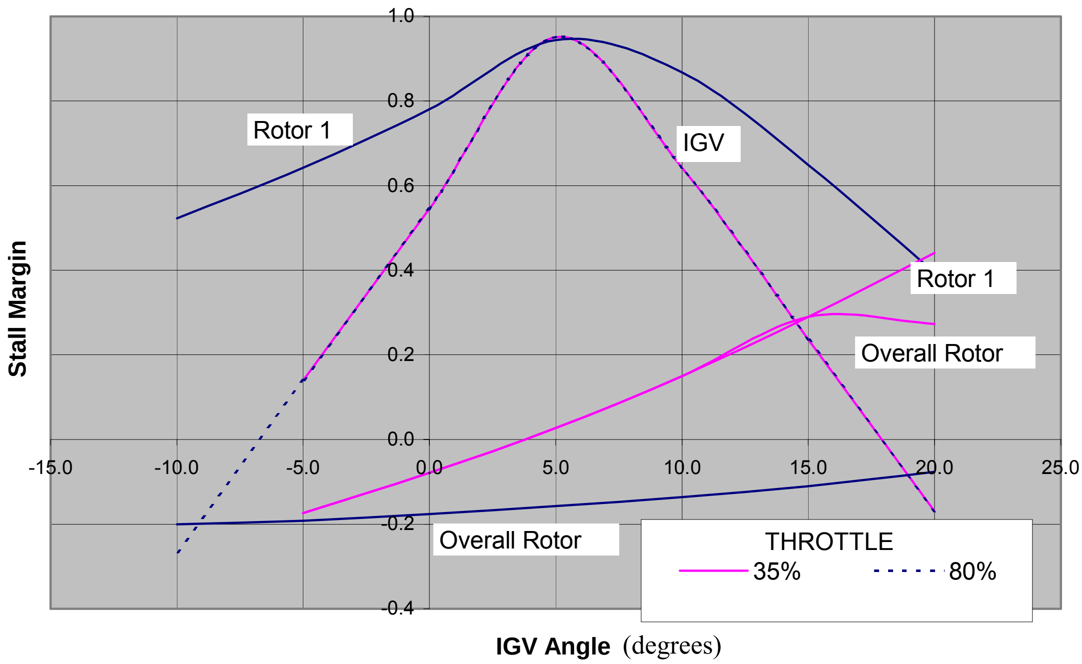
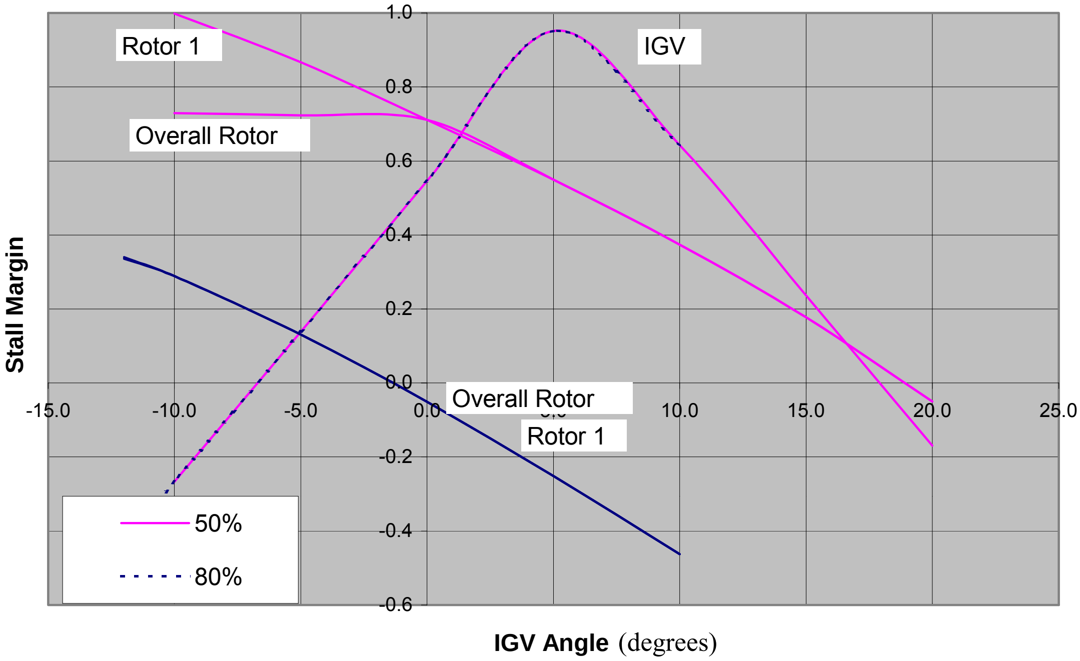


Fig 7.27 Effect on Stall Margins of Stator 0 (IGV) Adjustment

All other stators held at -10 degrees

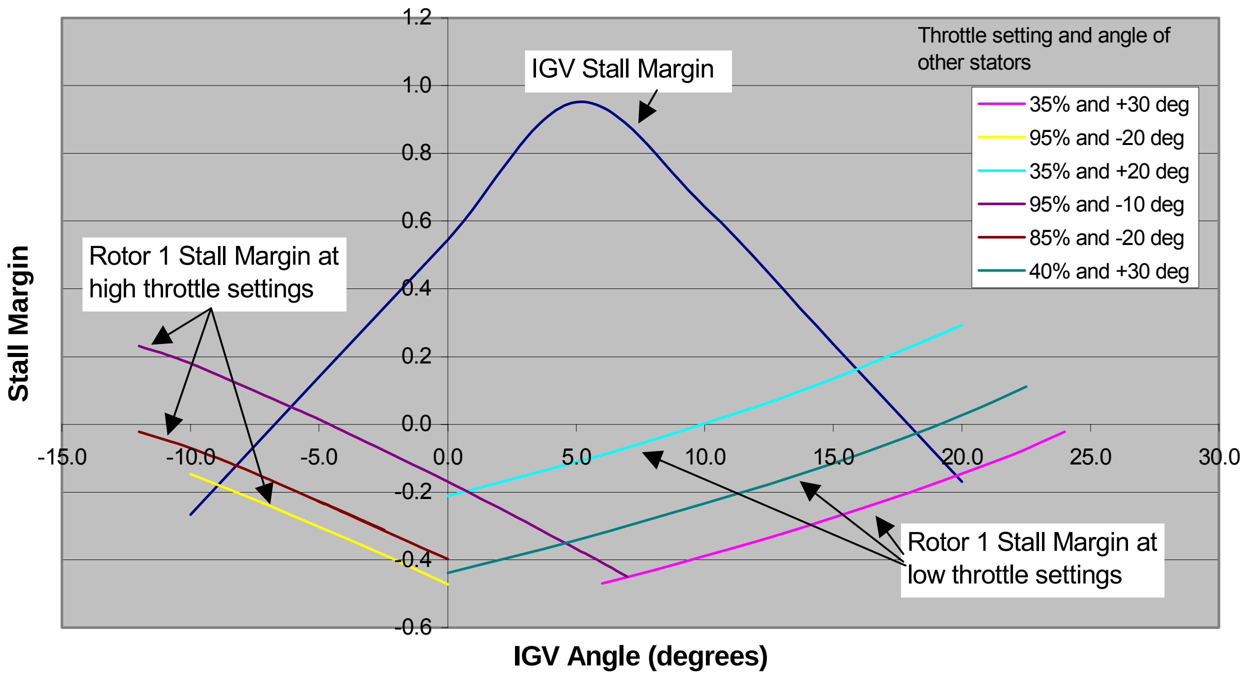


angle settings. In other respects the behaviour of the IGV is similar to that noted above for stator 3 i.e. strong effect on the stall margin of the downstream rotor row but relatively small effect on other rows as evident in the characteristic of overall rotor stall margin.

At high throttle setting with other stators at large negative angle, the rotor stall margin characteristic has shifted down towards the zero line and is of opposite sense to the adjacent characteristic for stator stall margin as shown in Fig 7.27. A similar effect is noted in Fig 7.26 at low throttle setting when other stators are at large positive angle. These observations suggest that conflict will arise if the intersection of the rotor and stator characteristics occurs at a point below the zero stall margin line. In such a case, adjustment of the IGV will produce an increase in stall margin of the stator but a decrease in that of the downstream rotor or vice versa. To demonstrate this point, further trials were run at more extreme settings and the resulting characteristics are shown in Fig 7.28 below.

Fig 7.28 Effect on Stall Margins of Stator 0 (IGV) Adjustment

Various settings of other stators



It is clear that conflict will occur at the more extreme settings represented in Fig 7.32. In practice, with priority given to stator stall margin control then this result means that the row agent controlling the IGV will not be able to control the stall margin of rotor 1 under such conditions and other agents will need to take the necessary corrective action.

For completeness, it is noted that stator 5 behaves very much like stator 3 with the obvious difference that there is no downstream rotor involved.

As expected, the observations from simulation confirm that for a large range of operating conditions the stall margin of a given rotor row is controllable by the setting of the upstream stator row. However, when the stall margins of a rotor row and upstream controlling stator row are both less than zero and their respective characteristics are of opposite sense then the control objective cannot be achieved. In such a case it is necessary to adjust all variable stator rows in order to bring about a change to the flow velocity vector (flow function) of the affected rotor row.

This result appears to be generally applicable to any stator and downstream rotor row combination. However, the IGV and rotor 1 represent a special case since the IGV stall margin characteristic is dependant only on the IGV angle and therefore the conditions under which the IGV is able to control the stall margin of rotor 1 are also suggested by the simulation results in terms of pressure and throttle settings.

Thus the rules for controlling rotor stall margin may be summarised as follows:

For stall margin of rotor row k+1 < 0:

Condition	Required Stator Action	
	β_k	$[\ \beta_i\ ]_{i \neq k}$
stator k stall margin changes in same sense as rotor k+1 stall margin OR stator k stall margin >>0	adjust stator k until rotor k+1 stall margin >0	no adjustment of other stator rows
stator k stall margin changes in opposite sense to rotor k+1 stall margin AND stator k stall margin ~ 0	no adjustment of stator k	adjust all stators except k until overall rotor stall margin > 0
in special case of IGV (k=0) with IGV stall margin ~ 0: at high pressure and low throttle setting OR at low pressure and high throttle setting *	adjust IGV until rotor 1 stall margin >0	no adjustment of other stator rows

- based on initial observations, 'high' and 'low' are relative to design point values

The rules for correcting rotor stall margin in the above cases are summarised as follows:

when correcting downstream rotor row stall margin:

$$\beta[j]_{n+1} = \beta[j]_n - \text{sign}(dsmr[j+1]_n) \,.\, \text{sign}(d\beta[j]_n) \,.\, (k.smr[j+1]_n - \delta)$$

when correcting 'other' rotor row stall margin:

$$\beta[j]_{n+1} = \beta[j]_n - \text{sign}(dsmro_n) \,.\, \text{sign}(d\beta[j]_n) \,.\, (k.smro_n - \delta)$$

where : smr[j+1] = stall margin of rotor row j+1
smro = overall rotor stall margin
and other terms are as previously defined.

In implementing these rules two points to note are:

1) Each row agent must make provision for correction of downstream rotor stall margin separately from that for contributing to the correction of 'other' rotor row stall margin.
2) A row agent only contributes to 'other' rotor row stall margin correction when it has received a message to do so.

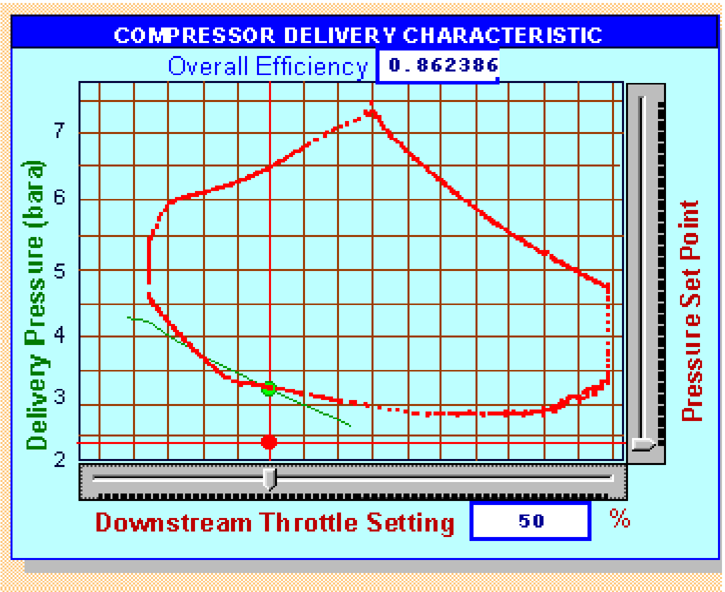
7.6.2 MAS Control

The program IGC2 was updated with the rules for rotor stall margin correction described in the previous section and then used to evaluate MAS control behaviour. The evaluation focused on operating range and efficiency optimisation.

a) Operating Range

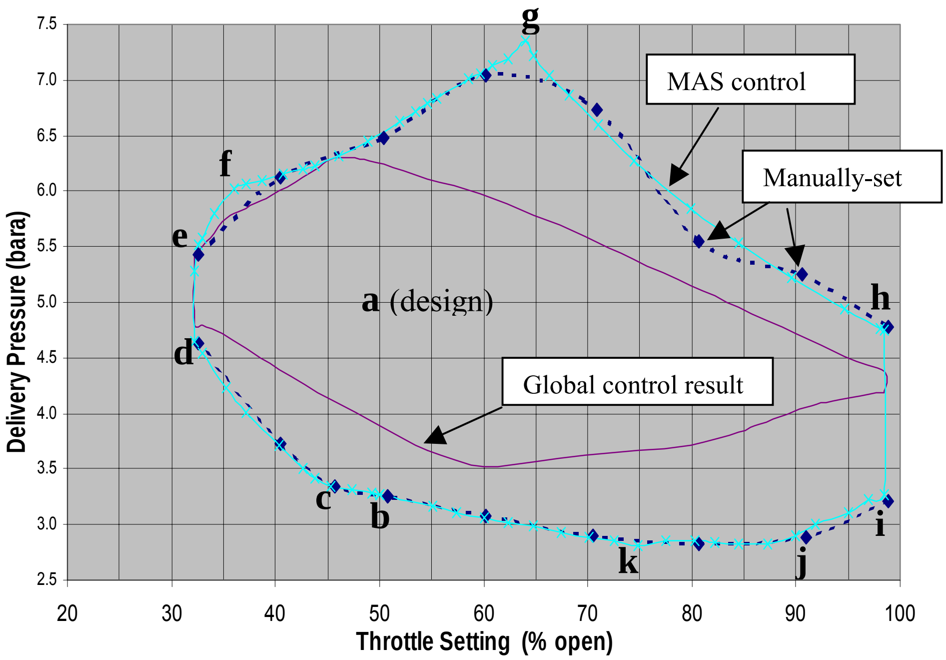
The first step in the evaluation process was to determine the operating range under MAS control. For this purpose all row agents are active and the efficiency agent is not selected. The trial was carried out by manipulating the user set point and throttle controls so that the target operating point moved around the extremes of the pressure - throttle field. The locus of the actual operating point achieved by the agents describes the boundary of the operating range. By definition, at points on the boundary the overall stall margin should have a minimal, but positive, value. As with IGC1, a feature was added to highlight operating points at which the overall stall margin lies between 0 and +5%. The overall result obtained after some development and refinement of the row agents' code is shown in the screen image of Fig 7.29.

Fig 7.29 Operating Range with MAS



The boundary data points were also captured and these are plotted in Fig 7.30 as a direct comparison with the operating range previously found by manual adjustment.

Fig 7.30 Comparison of Operating Range



The results indicate that the agent system is successful in extending machine operation to operating points, which are at, or close to, the limits imposed by stall margin. It is instructive to examine this result in detail and to relate system performance to the detailed operation of the row agents' computer code. To achieve this, the sequence of user control inputs is considered which cause the operating point to follow the path a-b-c-d-e-f-g-h-i-j-k shown on Fig 7.30. For each segment of this path, the operating conditions are described and the corresponding row agent code identified and explained. The relevant source code listing is given in Figs 7.31 and 7.32 for reference.

a* to *b

Set-point pressure change from design point to lowest pressure (~2.5 bara), throttle setting constant.

Fig 7.31 Row Agent Source Code - Stall Margin Correction

```
void rowagent::cognit()
{
  // *** STALL  MARGIN CORRECTIVE ACTION ***
A   //  Firstly respond to negative stator stall margin

    if(sms<0){
        ks=100*step;
        // adjust stator in direction of decreasing stator stall margin
        dbeta=-dirb*dirsm*(ks*sms-step);
        // set stall status flag
        stall=1;
        }


B   // Respond to downstream rotor row stall margin as long as this will not
    // cause negative stator stall margin

    else if(smr<0){
        kr=150*step;
        // if this row agent relates to IGV then apply special case rule
        if((row==0)&&(sms<0.01)){
            // adjust stator in pre-determined direction
            if((G<1)&&(pd>4.7))dbeta=-(kr*smr-step);
            else if((G>1)&&(pd<4.7))dbeta=2*kr*smr-step;
            else dbeta=0;
            }
        // otherwise apply general rule
        else if((sms<0.01)&&(dirsmr!=dirsm)) dbeta=0;
        // adjust stator in direction of decreasing stall margin of d/stream rotor
        else dbeta=-dirb*dirsmr*(kr*smr-step);
         // if this agent cannot make correction then request help from other agents
        if(dbeta = = 0) {
              for(i=0;i<6;i++){
                   if(i!=row)rowagent[i].put_message(row,0,1,1);
                   }
              }
         // set stall status flag
        else stall=2;
        }


C   // Respond to 'other' rotor row stall if requested to do so by other rowagent

     else if(smro<0){
        kr=50*step;
        if(rowagent_request==1){
       // adjust stator in direction of decreasing overall rotor stall margin
            dbeta=-dirb*dirsmo*(kr*smro-step);
            stall=3;
            rowagent_request=0;
            }
        // set stall status flag
        else dbeta=0;
        }
```

Fig 7.32 Row Agent Source Code - Delivery Pressure Control and Stall Avoidance

```
// *** DELIVERY PRESSURE CONTROL & STALL AVOIDANCE ***
D // Respond to set-point error while error outside tolerance and error change in progress
  else if((fabs(sperr)>tol)&&(dsperr!=0)&&(effagent_request==0)) {

      //  change stall status if sense of set-point error reversed
      if(dirp!=dirp0) stall=0;
      // or if corrective action for d/stream rotor is in same direction as set-point
      else if((dir!=dirp)&&(stall==2)) stall=0;

      // assign stall avoidance coefficent when at stall limits
      if(stall==1)fo=sms;
      else if(stall==2)fo=smr;
      else if(stall==3)fo=smro;

      // otherwise determine stall avoidance coefficient for set-point error change
      else if(stall==0){
              // downstream stall margin coefficient
              if(smr_next>0)fr=0.1;
              else if(smr>0.001)fr=smr;
              else fr=0;
              // stator stall margin coefficient
              if(sms_next>0) fs=0.1;
              else if(sms>0.001)fs=sms;
              else fs=0;
              // adopt lesser of stator and d/stream rotor stall margin coefficients
              if(fr<fs)fo=fr;
              else fo=fs;
              }

      // apply control law  (with stall avoidance coefficient) for stator adjustment
      kp=150*step;
      if(fabs(sperr)<1)dbeta=-kp*fo*sperr;
      else dbeta=-kp*fo*dirp;
      }
```

All row agents respond by executing code segment D, lines 7-18 (Fig 7.32) which causes the machine operating point to move towards the target set-point. During the transition, the stall margins of all rows decrease and those of the stator rows decrease the most. Stall margins are detected by the row agents and each determines a stall avoidance coefficient based on the lesser of the stator and downstream rotor row stall margins. If the expected next values of stall margin are positive then a constant is assigned to the avoidance coefficient.

If the set-point change is applied as a single step then it is noted that for some stator rows a momentary and small negative stall margin occurs before adjustment stops. This could be avoided if the rate of application of set-point change is reduced since the observation is not apparent if the set-point change is applied incrementally. At the end of the action, all stator stall margins are positive and close to zero.

b** to **c

Incremental reduction in throttle setting, set-point constant at minimum setting.

Each increment causes the operating point to move, momentarily, to a higher pressure point on the instantaneous delivery pressure vs. throttle setting characteristic. This causes the agents to respond in exactly the same way as for a set-point change as described above and returns the operating point as close to the target whilst retaining positive (near zero) values of stator stall margin. During the transition towards point ‘c’, the rotor stall margins continue to reduce with that of rotor row 1 (R1) having the lowest value.

c** to **d

Continued incremental reduction in throttle setting, set-point constant at minimum setting.

At point ‘c’ the stall margin of rotor row 1 reaches zero and further throttle reduction causes this to become negative. All agents detect this condition and execute stall margin correction code.

The row agent for the IGV executes lines 3 to 6 of code segment B (Fig 7.31) since it perceives that the associated downstream rotor row, R1, has a negative stall margin. (In the code *smr* = downstream rotor stall margin). Because the IGV stall margin is less than the specified limit and the operating point is outside of the range set in the 'special case' rule then the agent makes no change to the IGV angle. This is the correct response since at point ‘c’ a conflict occurs such that adjustment of the IGV in order to make the stall margin of R1 positive will result in a negative stall margin for the IGV. Since no correction can be made, the agent executes lines 9 to 13 of code segment B which sends messages to all other row agents requesting that they take action.

Other row agents at this time execute code segment C since they perceive R1 as 'other' than their associated downstream rotor row and react to the variable *smro*, overall rotor stall margin, rather than *smr*. In line 3 of the code, the agent checks if a request has been received from another row agent. If so, then line 4 is executed which causes adjustment of the local stator row in a direction to increase the variable *smro*. The process is repeated until the stall margin of R1 is positive. On completion, the agent sets a 'stall flag' to signify that corrective action has been taken.

Once R1 stall margin is positive, all agents resume the task of pressure control and execute code segment D. For the IGV row agent the stall flag has not been set since no corrective action occurred. This agent, therefore, executes from line 7 and determines a stall avoidance coefficient based on the lesser of the stall margins for the IGV and R1. Typically, since the IGV is at the stall limit then no adjustment will take place. The other agents will execute line 6 of Code D and set the avoidance coefficient equal to *smro,* the overall rotor stall margin, which by definition will be the same as that for R1. A very small adjustment towards the set point may result depending on the precise value of *smro.*

The net result of this simultaneous action by agents is that the machine characteristic, and hence operating point, moves to a higher pressure level. The process is repeated for each increment of throttle reduction until the model limit of 32% throttle setting is reached at point ‘d’. At this point, both IGV and R1 still have near-zero stall margins and the delivery pressure has risen to 4.64 bara.

d to e

Set point change from minimum to maximum pressure (2.5 to 7.3 bara), at constant throttle.

All agents respond by executing code segment D, line 2 of which causes the stall flag to be set to zero since a reversal of pressure error direction has been detected. Thus all agents execute code lines 7 to 15 as described for set-point change previously. During the transition the IGV stall margin increases while those of all other stator and rotor rows decrease until near zero at which point agent action ceases. Momentary excursions into corrective action of all types may be observed at point ‘e’ depending on the rate of application of set-point change.

e to f

Incremental increase in throttle setting with set-point constant at maximum value.

All agents respond by executing code D and apply the appropriate values of stall avoidance coefficient. In this region, the IGV stall margin decreases whilst those of the other stators remain at near-zero. Meanwhile, the stall margins of the rotor rows, with the exception of R1 which remains at near-zero, all increase. Eventually, at point ‘f’, IGV stall margin reaches zero and that of R1 starts to increase.

f to g

Continued incremental increase in throttle setting with set-point constant at maximum value.

As the throttle setting is increased further, the stator stall margins remain near-zero whilst those of the rotor rows increase, reach maximum values, and then decrease. As point ‘g’ is approached, the stator stall margins, except for IGV, begin to increase and that for R1 further reduces so that at point ‘g’ the limiting condition is set by the IGV and R1. During this transition from ‘f’ to ‘g’, the row agents have remained in pressure control /stall avoidance mode executing code D.

Point ‘g’ is the peak of the operating range boundary and coincides with the point at which the operating point is about to move from the positive to the negative slope of the pressure-throttle characteristic. Also at this point, the positive slope of the characteristic is almost vertical and therefore operation is close to being unstable. This condition is at the limit of the compressor flow model.

g to h

Continued incremental increase in throttle setting with set-point constant at maximum value.

This part of the operating range boundary mirrors section ‘c’ to ‘d’. The stall margin for R1 becomes negative as the throttle setting is increased and the agents respond accordingly.

The IGV row agent executes code B lines 3 to 6 resulting in no adjustment of the IGV and, consequently, lines 9 to 13 are executed which cause messages to be sent to other agents requesting action. The other agents execute code segment C, react to the messages received, and adjust their respective stators to restore R1 stall margin to a positive value. During this period, the stall margins of the stators, except IGV, increase, reach maxima, and then reduce. Eventually, at point ’h’, the IGV and R1 remain at stall limit with rotor 5 and stator 5 also approaching zero stall margin.

h to i

Set point change from maximum to minimum pressure (7.3 to 2.5 bara), at constant throttle.

This section of the operating range boundary mirrors section 'd' to 'e' and all agents respond to the set-point change by executing code D. During the transition all rotor row stall margins decrease whilst the IGV stall margin increases and other stator stall margins decrease. Thus at point 'i' all rotor rows are at the stall limit along with stator 5.

<u>i to j</u>

Incremental reduction in throttle setting, set-point constant at minimum setting.

The stall conditions at point 'i' remains through this section and the agents execute all types of corrective action to maintain positive values of stall margins for the rotors and stator 5. Meanwhile the stall margins of the IGV and other stators decrease so that when point 'j' is reached the IGV is at the stall limit as well as all rotors and stator 5. However, the stall margin for R1 is just about to increase.

<u>j to k</u>

Continued incremental reduction in throttle setting, set-point constant at minimum setting.

As the throttle setting is reduced there is a progressive transfer of limiting stall condition from the rotors to the stators, during which the agents continue to respond with appropriate corrective action to maintain a positive overall stall margin. When point 'k' is reached the transfer is complete and all stators including the IGV then have near-zero stall margins.

<u>k to b</u>

Continued incremental reduction in throttle setting, set-point constant at minimum setting.

This section is a continuation of section 'b' to 'c' and the stall conditions and agent response are thus as previously described.

If the sequence of user control input is reversed so that the operating range boundary is traversed in an anti-clockwise direction then the same basic contour is traced but with minor differences, particularly in section 'j' to 'i'. Differences arise because the agent response to throttle change depends on the direction of such change relative to the instantaneous operating characteristic. The response will be *corrective action* when the throttle change is in one direction and *stall avoidance* when in the opposite direction. The respective codes are not identical and hence the results achieved for the same throttle setting are slightly different. Also the contour shown in Figs 7.29 and 7.30 is strictly only obtained when the throttle changes are small. Large changes in the throttle setting produce significant excursions into negative stall margin before the agents react. This is a feature of the simulation model, as previously explained, and can produce a 'saw-tooth' effect on the contour.

It was noted that at operating points within the operating range boundary the overall stall margin may be close to zero on occasions. This is to be expected since the control strategy requires the row agents only to avoid (or, if necessary, correct) negative stall margin when responding to set point error. No facility is provided to control the <u>relative</u> position of stators during such response. Consequently, the resulting stator configuration i.e. pattern of stator row angles depends on the history of previous adjustments and is unlikely to be optimal for any given operating point. For this reason the strategy includes an efficiency agent whose purpose is to optimise the stator configuration once a stable operating point has been achieved by the row agents.

<u>b) Maximising Efficiency</u>

The program code for the relevant routines finally implemented in the efficiency agent and row agent is shown in Figs 7.33 and 7.34 respectively. The basic operation of this code is described as follows.

When row agents complete set-point response they enter "no control action" state, code E (Fig 7.34) lines 18 to 22, and communicate this status to the efficiency agent using *put_message( )* method. When the efficiency agent registers that all row agents are in this inactive state then a 'new operating point' flag is set, code G (Fig 7.33) line 7, and the maximise efficiency routine commences, code G lines 10 to 21.

The efficiency agent starts the process by sending a message to row agent 5 requesting that agent to “maximise efficiency”, code G line 14. It will also save the current value of overall efficiency at this time for reference. The efficiency agent, effectively, then does nothing until a message is received from row agent 5 indicating that maximum efficiency has been achieved, code G line 15. The next upstream row agent is selected and the efficiency agent sends the “maximise efficiency” request as before. After all row agents have responded, the efficiency agent checks the current overall efficiency against the reference value (code G line 19) and if any reduction in efficiency has been detected during the past round of adjustment then a counter is incremented. When this counter exceeds a pre-set value (currently 5) the 'efficiency at maximum' flag (eff_max) is set and the process halted, otherwise a new round of adjustment is initiated.

On receiving a request from the efficiency agent, a row agent executes code E (fig 7.34) lines 1 to 16. The first action is to determine the direction in which to adjust the stator to increase efficiency, code E lines 3 to 11. After that, the stator is simply incremented in steps (1 step = 0.1 degree) as long as an increase in efficiency is detected. If the set point error is outside tolerance then no stator adjustment is made (E line 12).

Whilst a selected row agent is responding to the efficiency agent request, the other row agents are executing code D of Fig 7.32 in order to maintain the current set point. Should a change in operating point be detected by the efficiency agent (code G line 3) due to a throttle or pressure set point change then the 'efficiency at maximum' and 'new operating point' flags are cleared and the current optimisation is disengaged.

Fig 7.33 Efficiency Agent Source Code – Maximise Efficiency

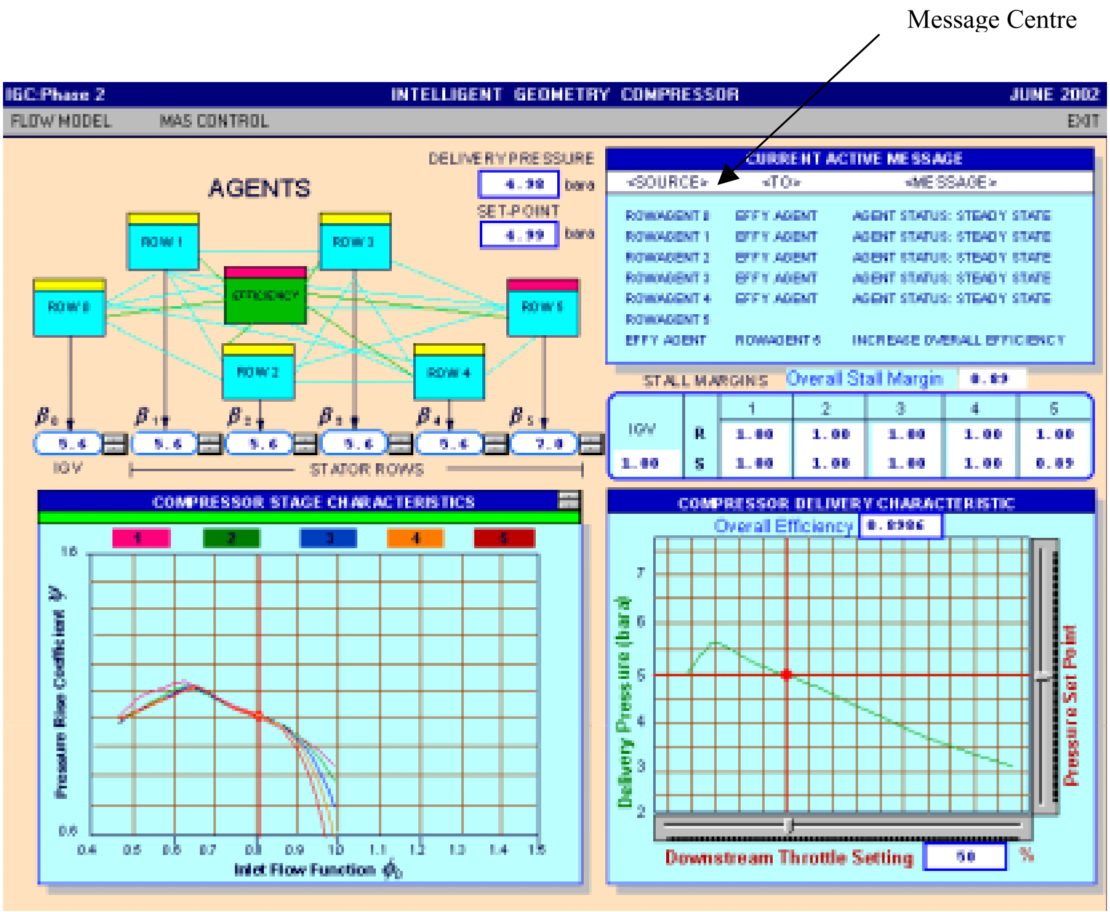
```
void effagent::cognit()
{
G  //Determine status of row agent from received messages. Assign value to flag to reflect row agent status:
 //row status =0 signifies all row agents in steady state
 // row status =1 signifies at least one row agent not in steady state
 row_status=0;
  for (i=0;i<6;i++) if (agent_stat[i]!=1) row_status=1;

 //Reset control flags if change in target operating point detected i.e. a change in throttle setting point
 // or pressure set-point and disengage efficiency maximising process
   if((fabs(dG)>0.001)||(fabs(dpset)>tol)){
        eff_max=0;
        new_opoint=0;
        }

   //otherwise if  operating point change is in progress, watch for new operating point and assign flag when reached
    else if((eff_max==0)&&(new_opoint==0)){
        if(row_status==0) new_opoint=1;
         row=5;   // reset first row in cycle
         cycle=0; // reset cycle counter
        }

   //otherwise when  new operating point reached then implement process to maximise efficiency
   else if((eff_max==0)&&(new_opoint==1)){
        if(start==1){
                //if at start of round of adjustment then save current efficiency for reference
                if(row==5) oeref=oe;
                start=0; // clear start flag
                }
        //send message to selected row agent to move to point of max efficiency
        rowagent[row].put_message(6,0,4,0);

        //if row agent at maximum efficiency (status=5) select next row or stop
        if(agent_stat[row]= =2){
                //select next upstream row
                if(row>0)row=row-1;
                //if all rows adjusted then select downstream row (5)
                else{
                       row=5;
                       //  if any significant reduction in efficiency detected during last round of adjustment
                       //  then increment cycle counter
                       if((oe - oeref) < 0.0001) cycle = cycle + 1;
                       //  if cycle count exceeds 5 then set control flag to stop process
                       if ( cycle > 5) eff_max=1;
                      }
                start=1; // set start flag
               }
         }

  //No control action
  else status=0;

} // end effagent::cognit
```

Fig 7.34 Row Agent Source Code – Maximise Efficiency

```
    // *** DELIVERY PRESSURE CONTROL & STALL AVOIDANCE ***

D    // Respond to setpoint error while error outside tolerance and error change in progress
1      else if((fabs(sperr)>tol)&&(dsperr!=0)&&(effagent_request==0)){

       LINES 1 TO 18 (Fig 7.32)

E      // *** EFFICIENCY ***

       // If maximise efficiency request received then find point of maximum efficiency for current throttle setting
1      else if(effagent_request==1){

               // set row status to signify efficiency routine in progress
2              rowstat[row]=5;
               // on first cycle of routine set references & make trial adjustment
3              if(cycle==1){
4                      dbeta=direff*step;
5                      cycle=2;
                       }
               // on second cycle of routine reverse direction of adjustment  if reduction in efficiency
               // from cycle 1 trial was detected
6              else if(cycle==2){
7                  if(doe<0){
8                      direff=-direff;
9                      dbeta=direff*step;
10                     }
11                 cycle=0;
                   }

               // on subsequent cycles continue to adjust in direction of increasing efficiency as long as
               // set-point in tolerance
12             else if(fabs(sperr)>tol) dbeta=0;
13             else if((dbeta!=0)&&(doe>0))dbeta=direff*step;
               // otherwise quit process and send message to efficiency agent
14             else{
                   // pass message to efficiency agent to signify end of process
15                 effagent.put_message(row,0,3,5);
16                 cycle=1; // reset for next time
                   }
               }

          // No control action
18     else {
           rowstat[row]=0;
19
20         dbeta=0;
21         cycle=1;
           // pass current status to efficiency agent
22         effagent.put_message(row,0,2,rowstat[row]);
           }

     // Set stator row angle
       CODE OMITTED
} // end rowagent:: cognit
```

Whilst the efficiency maximising process is in progress the message exchange between the row agents and the efficiency agent is displayed in the table on the simulation window as the screen image below indicates.

Fig 7.35 Simulation Display During Efficiency Routine



In general, the degree of improvement in efficiency that is achieved at a given operating point depends on the stator row configuration arrived at by the preceding row agent action. In Fig 7.36, the results obtained when the stator rows are initially at their design

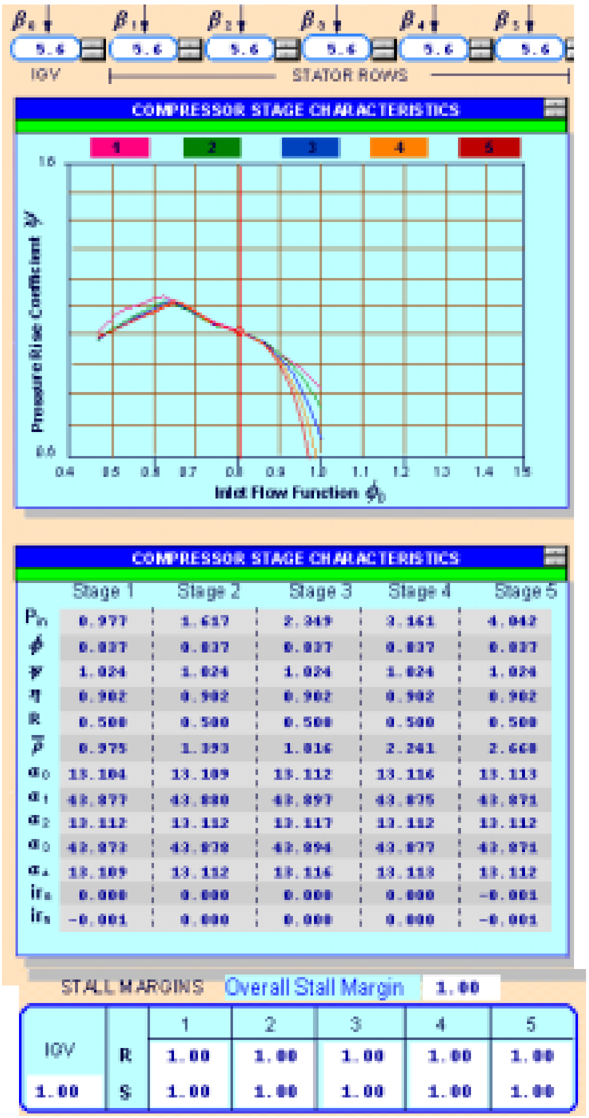
Fig 7.36 Efficiency Maximising at Design Point

Delivery Pressure = 4.99 bara

Throttle Setting = 50%

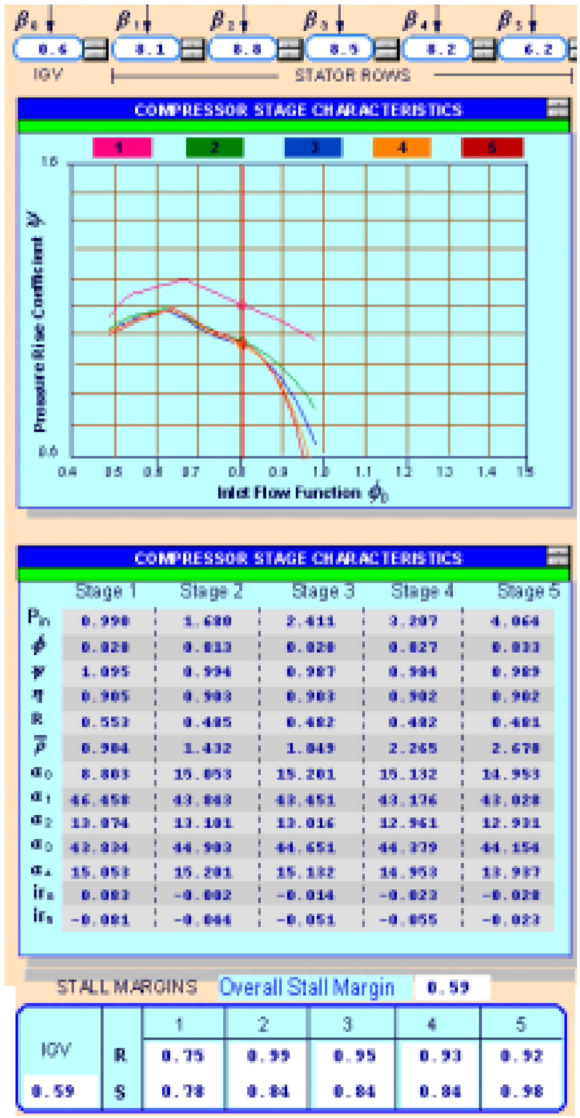
Before

Overall Efficiency = 0.8985



After

Overall Efficiency = 0.8997



settings, i.e. all at 5.6 degrees, are shown. After the optimisation process the following differences in performance parameters are noted:

a) overall efficiency has increased by just 0.1%
b) IGV stall margin has reduced from 1.0 to 0.59
c) pressure drop across IGV reduced
d) stage 1 efficiency has increased by 0.3%
e) stage 2 and stage 3 efficiency has increased by 0.1%
f) stage 4 and stage 5 efficiency is unchanged
g) stage 1 (Ψ-Φ_0) characteristic has separated from those of the other stages
h) IGV angle has reduced by 5 deg whilst all other stator row angles have increased

Similar observations were apparent at other, randomly selected, operating points over the operating range with little or no increase in overall efficiency being detected. It appeared that, contrary to expectations, the MAS control action results in stator settings, which are generally close to optimal values for maximum efficiency.

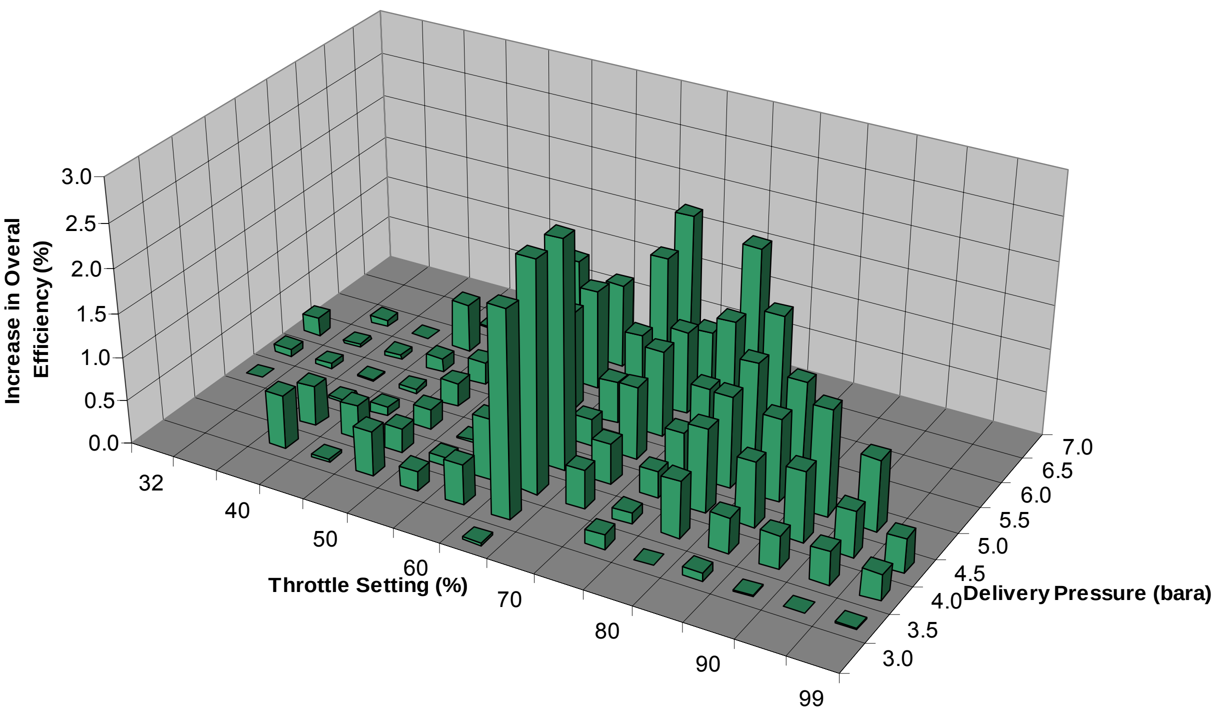
A more formal trial was conducted in which the full operating range was traversed in two successive runs. In both cases, values of efficiency were recorded at each of a number of points over the maximum set-point range for different throttle settings. This ensured that the operating point was forced to the boundary of the operating range for each throttle setting. On the first run, optimisation was disabled. On the second run, which used the same operating points as the first run, optimisation was enabled. The results for the two runs were compared to reveal changes in efficiency at each of the operating points included in the trial. The outcome is shown in the 3-D chart of Fig 7.37, which plots efficiency change over the pressure-throttle field.

The trial revealed significant increases in overall efficiency at the majority of the operating points tested. This suggests two conclusions:

1) MAS action at operating points on the boundary of the operating range can lead, subsequently, to sub-optimal settings of stators.

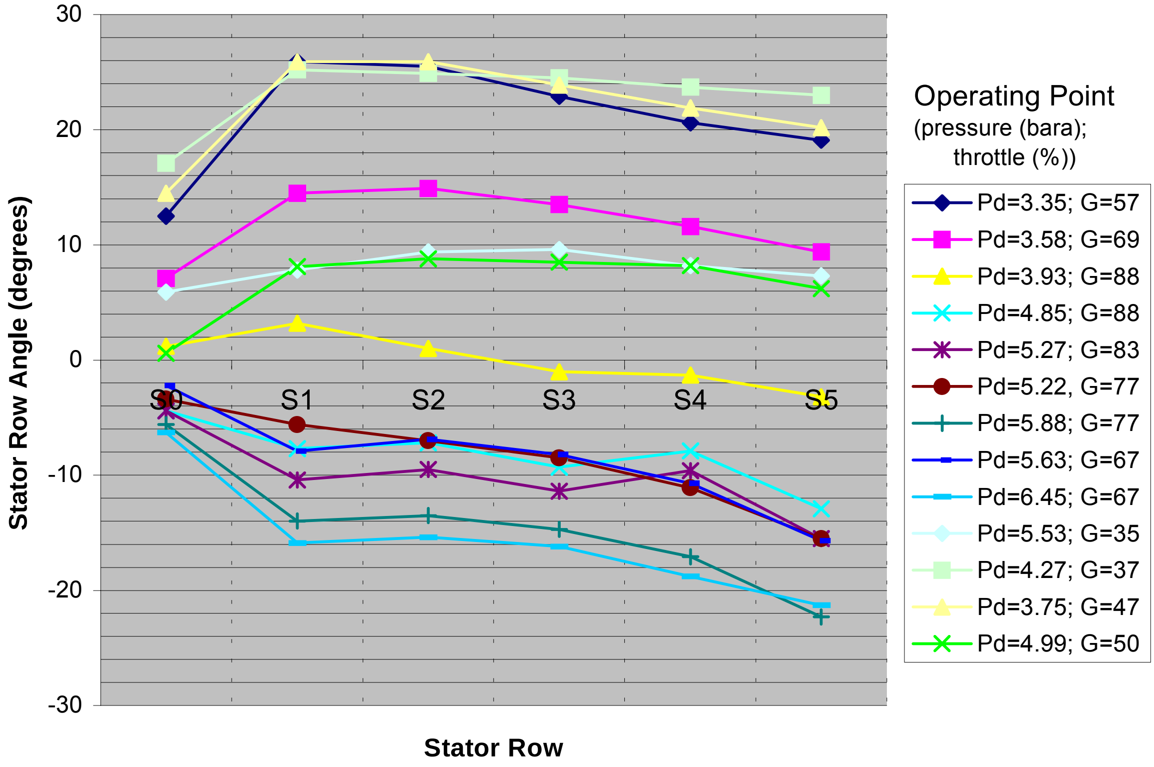
2) The optimisation process improves initially sub-optimal stator settings to recover values of efficiency that might usually be achieved if MAS action is confined within the operating boundary.

Fig 7.37 Optimisation at Selected Operating Points



The stator configurations that result from the optimisation process were investigated in a further trial and the results are shown in Fig 7.38.

Fig 7.38 Stator Configurations after Optimisation



With the exception of the IGV, these results suggest that an optimised stator configuration for operating points away from design is one in which stator angle progressively decreases for rows further downstream. At or close to design, an equal angle configuration appears to be optimal.

7.7 Conclusions

Phase 2 of the research demonstrated, through simulation, a multi-agent system (MAS) for controlling the steady-state performance of a variable geometry axial flow compressor. The MAS comprises a set of *row agents*, one for each variable stator row, and an *efficiency agent.* The agents are *reactive* entities, each responding to changes in the common fluid environment. The row agents act independently to achieve the overall system goals of delivery pressure control and stall avoidance. However, row agent action does not necessarily produce stator configurations, which are optimised for overall efficiency. For this purpose, the efficiency agent co-operates directly with row agents, using point-to-point communication and message-passing, to modify the stator configuration and thereby improve overall efficiency.

The row agents operate concurrently and use simple rules to determine the angular adjustment of an associated stator row. Although operating independently, row agents co-operate through message passing when reacting to conflicting rotor-stator stall situations. However, to achieve satisfactory co-operation in this way it was necessary to build in 'special case' rules which are application-specific and thus contrary to the desirable goal of universality in the control strategy and methods. Also, it was necessary for agents to have information about the overall performance of the machine as well as local information to achieve their objectives. In practice, it may prove very difficult to provide such information to all agents at one time, consistently, with the result that system performance may be impaired. (It may be noted that in the simulation program the underlying thread synchronisation ensures consistency of 'sensor' data for all agents i.e. all agents see the same data values at the same time.)

Notwithstanding the above reservations, the simulation results show the implemented multi-agent system to be very effective in achieving the system goals and enabling a much extended operating range to be achieved compared to that of the datum system described in Phase 1.

The ideal boundary of the operating range for the machine is defined in this research to be formed by those operating points at which stator and rotor stall margins are zero. Predicting such a boundary, mathematically, would not be a trivial task. Yet by applying simple rules to the information sensed from the environment, the row agents are able to determine such a boundary which, as the simulation results show, is of a quite complex form. In this way the MAS demonstrates that parallel processing of simple rules offers an effective alternative to a more conventional approach based on a complex mathematical model. It should also be noted that in all the simulation runs so far, MAS action always converged to a stable operating point. Intuitively, with many separate entities interacting in a common environment there would seem to be risk of control system instability.

Simulation results showed the effectiveness of the optimisation process in improving sub-optimal stator settings to achieve increases in overall efficiency whilst maintaining a given operating point. However, the occurrence of sub-optimal settings was noted only after repeated excursions to the extremes of the operating range. When operation was confined within the regulation boundary the relative stator settings arrived at by independent row agent action appeared to be close to optimal and allowed little scope for improvement.

The stator settings produced by the optimisation process appeared to conform to a regular pattern but it is not possible to draw any general conclusion from this observation at this stage. For reference, it is noted that Riess and Blöcker (1987) investigated three patterns (or schedules) of stator settings in their experimental variable geometry machine. These were equal angle, linear increasing angle downstream and linear decreasing angle downstream.

CHAPTER 8

IGC Design Phase 3

Multiple Agents and Independent Control of Stator Vanes

8.1 Objectives

The multi-agent system developed in the previous phase of the research was based on the control of individual variable stator rows. In this phase, consideration was given to the control of individual vanes within each stator row in order to demonstrate increased flexibility of control over machine performance. Also this provided experience of a system involving a much larger number of agents.

As explained in chapter 6, the limit on steady-state operating point for the hypothetical compressor is defined in terms of (static) stall margin. Thus, in the case of stator row control, the limiting condition for the machine was set by the lesser of the (mean) stall margins of all stator and rotor rows. In this phase of work, this idea was extended such that the limit on machine operating point is now set by the lesser of the stall margins of all stator vanes and rotor rows. Thus, by this definition, if the stall margins of individual vanes within a given row differ then the overall stall margin, and hence the operating range of the machine, will be affected. By controlling the angular setting of each stator vane independently the MAS seeks to counter local flow effects and thus maximise the overall operating range.

In implementing the vane agent system, the opportunity was taken to develop an alternative approach to the optimisation of stator geometry. Previously, the objective of maximising efficiency at a particular operating point was achieved by successive adjustment of each stator row in turn whilst monitoring a control variable which directly represented overall efficiency. The alternative approach was based on the previous observation that at a given operating point, greater efficiency appears to correlate with greater stall margin. Therefore, the MAS seeks to maximise overall stall margin at a particular operating point as an indirect way of increasing efficiency.

It was necessary to modify the simulation program to support the demonstration and evaluation of the extended MAS with independent stator vane control. The revised program was designated IGC3.

8.2 Total System Model

This remains as previously defined, comprising the IGC operating at constant rotational speed with the downstream load represented by a variable throttle valve. Overall performance is defined by delivery pressure, mass flow rate, efficiency and overall stall margin. All of these variables were assumed to be measurable and available for purposes of control. A set-point value for delivery pressure is provided from an, unspecified, external source.

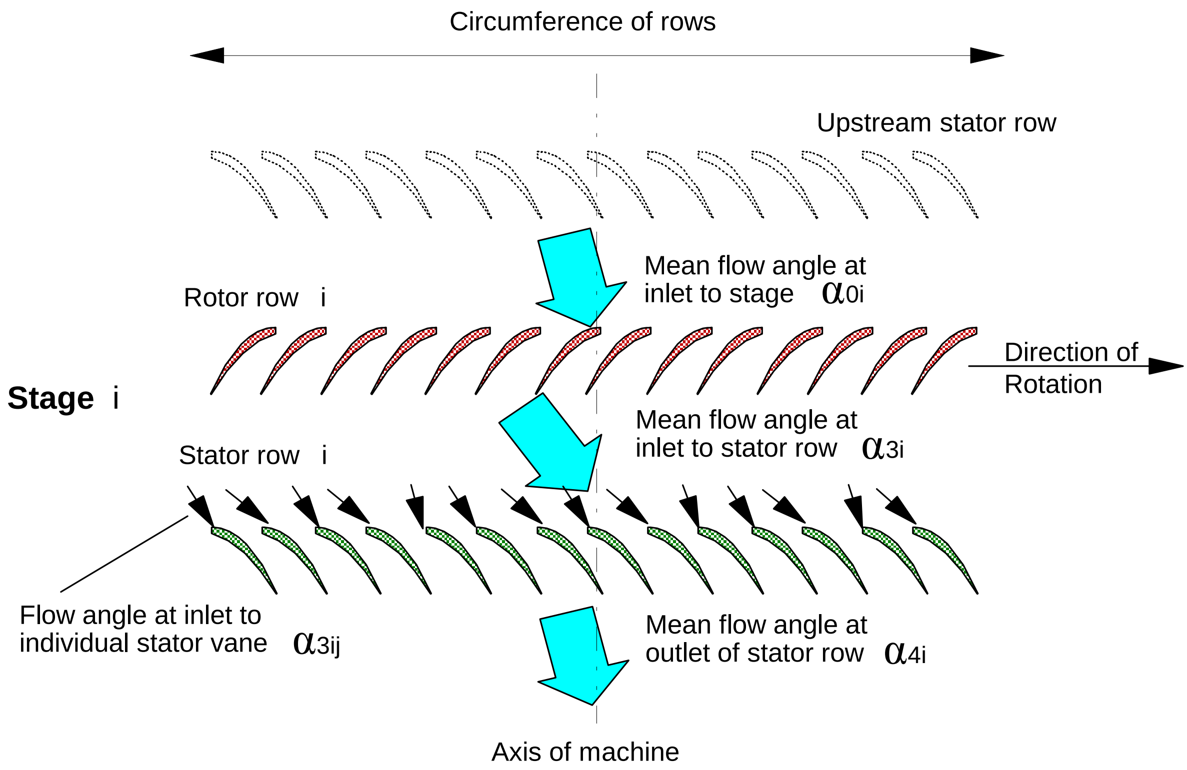
8.3 Flow Model

The flow model used in the simulation so far is based on the calculation of steady-state characteristics for each compression stage of the machine. In this, it was assumed that flow is steady and axi-symmetric and that flow variables represent mean values across the flow stream. As such, the model does not provide any mechanism for reflecting the behaviour of an individual variable stator vane. The model therefore needed to be extended to allow different flow conditions at each vane within a row. This was achieved by the introduction of a circumferential flow variation within the annulus between adjacent rotor and stator rows.

In order to minimise the amount of change to the flow model and the related computational modules and also to retain direct comparability with the results of the previous work an approximate representation of flow variation was adopted.

As before, the characteristics for a given stage i are calculated based on the mean flow angles at entry and exit to the rotor and stator rows. However, superimposed on the mean flow is a circumferential variation in axial flow velocity in the annulus between the rotor and stator. This flow variation is represented in the model as a variation in the absolute inlet flow angle at each stator vane as shown in the diagram of Fig 8.1.

Fig 8.1 Flow Model for Stage



Note: the flow angle notation follows that given in the Appendix.

The inlet flow angle to the j^{th} vane in stator row i may be expressed as:

$$\alpha_{3ij} = \alpha_{3i} + \delta\alpha_{3ij}$$

where α_{3i} = mean inlet flow angle to stator row i

and $\delta\alpha_{3ij}$ = variation in inlet flow angle

By adopting values of $\delta\alpha_{3ij}$ within the constraint that:

$$\sum_{j=1}^{j=m_i} \delta\alpha_{3ij} = 0$$

the validity of the mean calculation is maintained. However for any particular vane the calculated value of incidence and, hence stall margin, will be modified by the amount of variation $\delta\alpha_{3ij}$.

The calculation of the mean flow angle at exit from stator row i, $\alpha_{4i,}$ is based on correlation data, the particular values of which depend on the mean stagger angle of the stator row β_i.

Thus:

$$\alpha_{4i} = \alpha_{3i} - f(\alpha_{4i} - \beta_i)$$

where the mean stagger angle is taken as the mean of all vanes in the row

i.e. $\beta_i = \sum_{j=1}^{j=m_i} \beta_{3ij}$

In this way, adjustment of an individual stator vane will be reflected in the mean angle of the stator, which, in turn, will affect the mean stage characteristics. For any stator vane j, the change in incidence from the mean of the row will simply be the net effect of the variation in inlet flow angle and the adjustment of vane stagger angle.

Thus:

$$\delta(incidence)_{ij} = \delta\alpha_{3ij} - (\beta_{ij} - \beta_i)$$

From the above, the model now provides, on the one hand, a means of introducing a circumferential flow variation through the variation in flow angle and, on the other hand, the means of countering the effect through adjustment of individual stator vane.

In application, the variation in flow angle at each stator vane is re-calculated after a change in machine operating point. The value of the variation is randomly set to be a small proportion of the current mean value of inlet flow angle within a range $\pm k\alpha_{3i}$ where k is a constant e.g. 0.01. In the case of the IGV where the inlet flow angle has been assumed to be zero always, then the variation is set within an absolute range of typically ± 2 degrees.

The number of vanes in each stator row was calculated from the geometry of the vane and the dimensions of the annulus. The detail of the calculation is given in the Appendix from which the results are as follows:

stator row	number of vanes
0 (IGV)	16
1	16
2	17
3	18
4	19
5	20

The total number of stator vanes is 106. This relatively small number is a consequence of the large dimensions chosen for the hypothetical machine.

It should be stressed that the above method is not proposed to be an accurate representation of a real flow situation but is simply an expedient way in which to introduce effects into the compressor flow model which have significance at the level of an individual stator vane. However, in a real compressor, it would be expected that local flow conditions would vary from vane to vane due to the 3-dimensional and turbulent nature of the flow and also due to small differences in the form and condition of each vane. Therefore the approach adopted is not without some relevance to a real machine application.

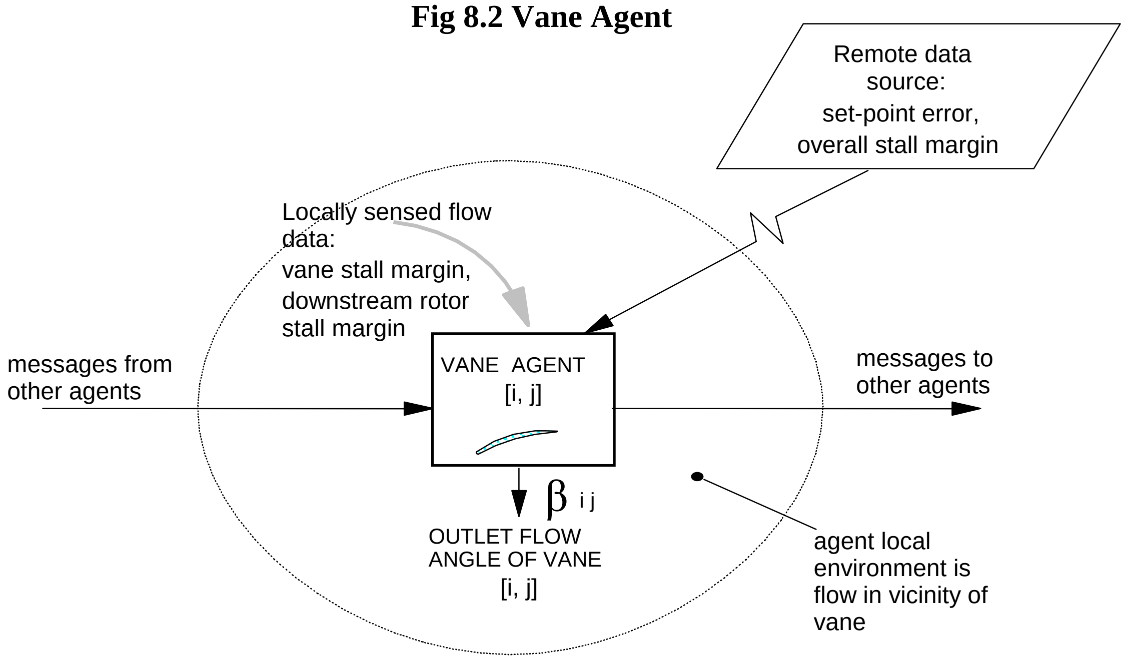
8.4 MAS Design

The multi-agent system for controlling stator vanes is a direct adaptation of that developed in the previous phase of work for controlling stator rows. It differs mainly in the scale of the system i.e. in the number of agents involved and the scope of their respective local environments. Overall system operation is the same as previously described.

8.4.1 Constituent Agents

Effectively, each row agent of the previous MAS was decomposed into a set of 'vane' agents, one relating to each vane in a stator row. Thus the system comprises, in total, 106 ***reactive autonomous*** agents. The functional scope of a vane agent is similar to that of the row agent but with the primary goal being stall avoidance/correction of the related vane rather than of a stator row. In respect of the secondary objectives of responding to set-point error, downstream rotor stall margin and overall stall margin, the vane agent behaves exactly as the row agent. As before, optimisation of stator settings takes place once the set point has been reached and conditions are stable. The method adopted (described later) is carried out by the vane agents and does not require the services of a separate ‘facilitator’ agent. The vane agent is represented conceptually as shown below.

Fig 8.2 Vane Agent



8.4.2 Agent Interaction

A vane agent may potentially interact i.e. send/receive messages with any other vane agent. However, for the objectives being investigated here the need for interaction arises only in the situation where the correction of vane stall margin is in conflict with correction of downstream rotor stall margin. In this case, based on the experience of the previous MAS, the vane agent broadcasts a message to the vane agents of all other stator rows requesting that they co-operate in the corrective action.

Since all vane agents in the affected stator row act in the same way then the result will be that the agents of other rows will receive a great number of redundant messages. In practice, this may be an undesirable situation and can be avoided either by retaining 'row agents' for handling inter-row communication or, more simply, by nominating one vane agent in each stator row to be the 'transmitter' of any messages destined for agents in other rows. For simulation purposes redundancy is not a problem and therefore all vane agents are allowed to broadcast.

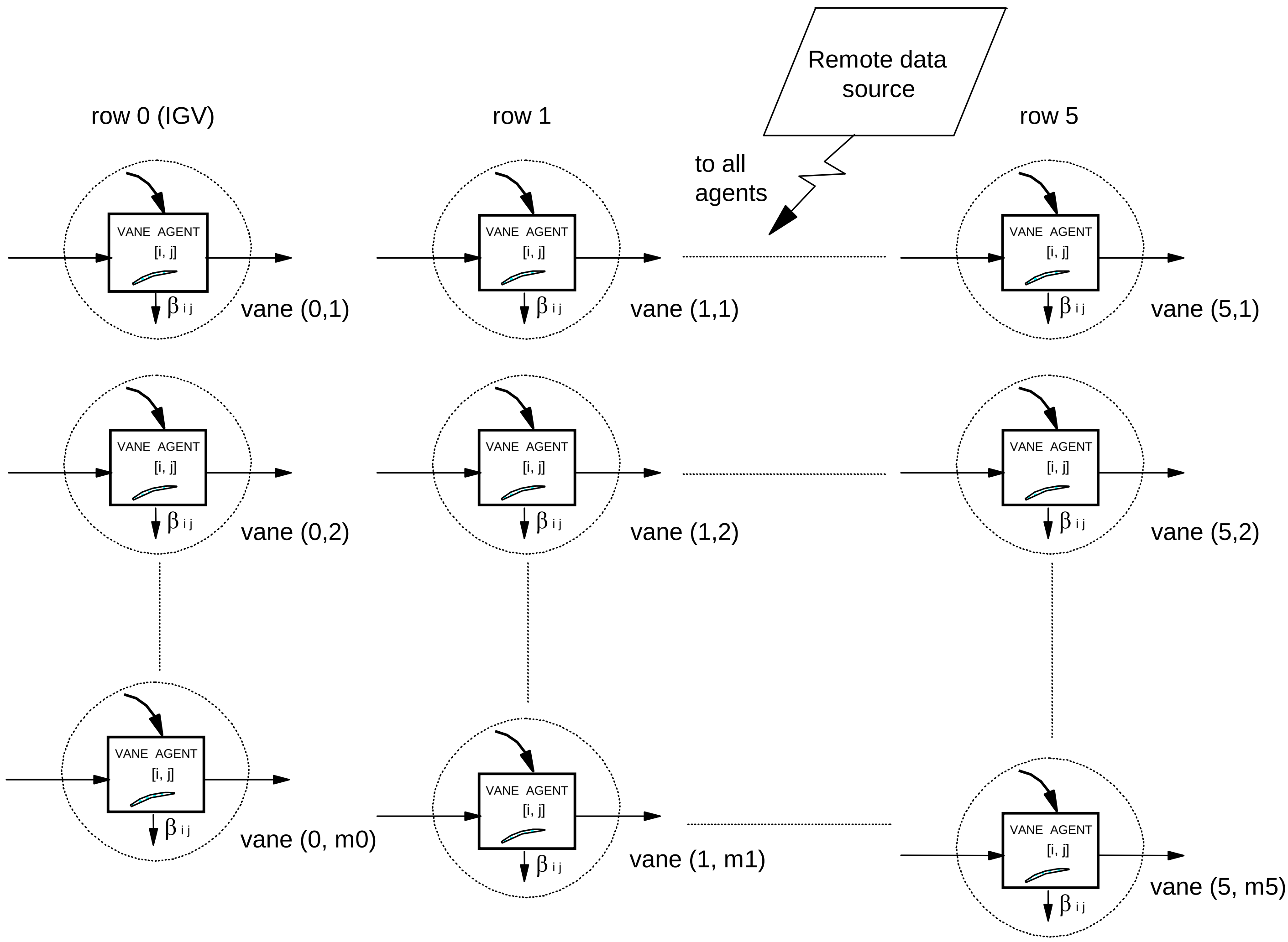
8.4.3 MAS Architecture

The system is essentially a collection of similar entities reacting autonomously to events in their respective local environments and to common information (from some unspecified source) which represents the overall state of the machine.

The form of interconnection between agents may be generally regarded as an agent network or, if confined to the minimum necessary to support the degree of co-operation described above, may be more restricted and could be configured as a combination of sub-networks. This point is of greater significance for real system implementation rather than for simulation. It is, however, useful to have some conceptual representation of system architecture and it is probably most meaningful to base this on the physical distribution of agents. This is shown in Fig 8.3 in which the system is represented as an array of vane agents each corresponding to the location of individual stator vanes.

Fig 8.3 System Architecture

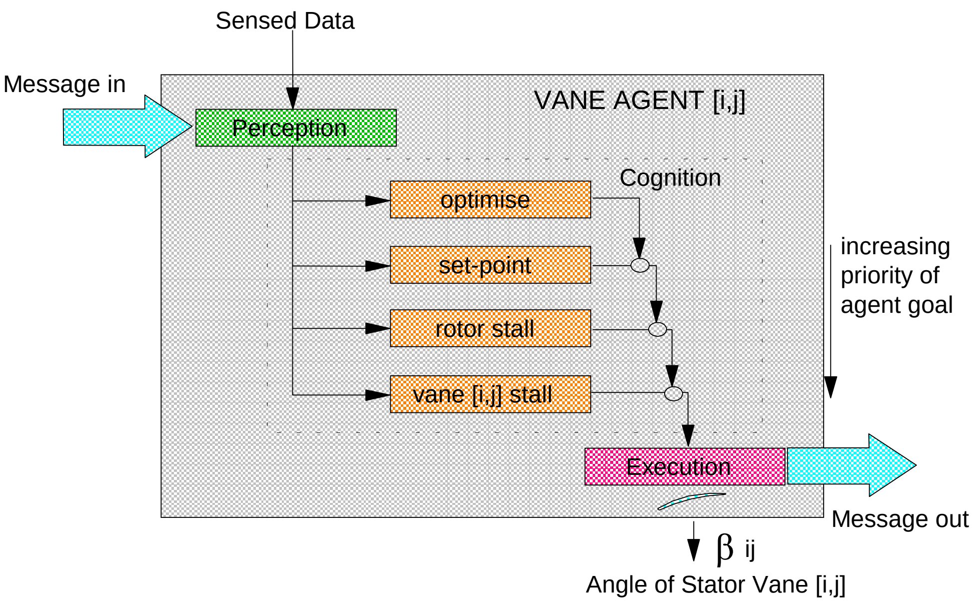
Network of Vaneagents - Interconnections omitted



8.4.4 Agent Design

The internal structure of the vane agent is identical to that of the row agent of the previous MAS with functional modules corresponding to prioritised agent goals. The architecture of the agent is shown below.

Fig 8.4 Vane Agent Architecture



The specific control algorithms, i.e. methods, employed in the modules were, with the exception of 'optimise', derived directly from those used in the corresponding row agent modules of the previous MAS with only minor change to extend control action to individual vanes.

The MAS strategy for optimisation of stator angles was based on the following rationale. The objective is to maximise overall efficiency whilst maintaining a given operating point.

Overall efficiency is defined in this research as:

$$\eta_{o/a} = \frac{\Delta_A}{\Delta_A + \Delta_L}$$

where: Δ_A = actual pressure **rise** from machine inlet to outlet

and Δ_L = actual pressure **loss** from machine inlet to outlet

Thus if the operating point is held constant then an increase in efficiency is achievable only by reducing the pressure loss. In the flow model used here, a significant component of pressure loss is related to the incidence (relative to design values) of stator vanes and rotor blades via the empirical correlation data of Howell and Bonham (1950) (ref. Appendix 1). Thus at many operating points it is expected that a decrease in incidence at a particular stator row will give rise to a net reduction in pressure loss. (A contrary result was observed in the previous phase concerning the IGV at design but this is considered to be exceptional.)

Therefore it follows that an increase in overall stall margin would, in general, be expected to produce an increase in overall efficiency. In fact, maximising stall margin is what the agents do when an operating point is beyond the boundary set by the limit of zero stall margin. Thus, if the limiting value for stall margin is set to be just greater than the overall stall margin associated with a required operating point then the agents will automatically maximise the overall stall margin as they seek to maintain the operating point. The proposed strategy for stator angle optimisation is therefore:

a) firstly, with stall margin limit set to zero, achieve required operating point
b) increase stall margin limit incrementally whilst maintaining operating point
c) repeat b) until required operating point cannot be achieved (by small amount)

If successful, the overall stall margin will be maximised and the actual operating point will closely approximate the required set point.

8.5 Simulation Program

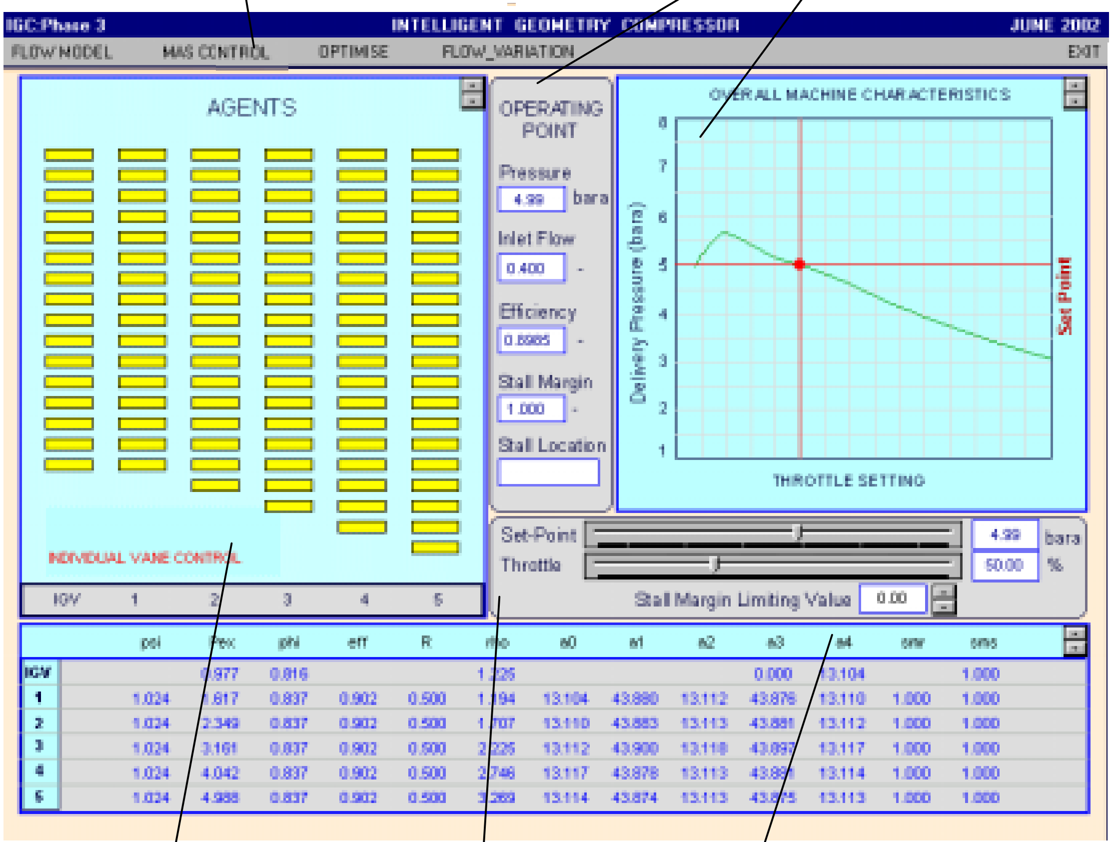
A third version of the simulation program, IGC3, was created to aid the evaluation of the multi-agent system for stator vane control. The program structure of the earlier versions was retained, as were the basic computation modules for flow. However, significant changes were necessary to provide the large number of agents and the related data sets. The main changes are described below.

8.5.1 Main Program and User Interface (Module: *IGC3main*)

The simulation window was re-designed to enable the states of all vane agents and all computed results to be displayed. The display comprises six parts as indicated in the screen image below.

b) operating point panel

a) main menu bar **Fig 8.5 Phase 3 Simulation Window** c) graphs window



d) MAS window e) user controls f) computed data tables

a) main menu bar

FLOW MODEL - starts and stops flow model

MAS CONTROL -

ROW CONTROL

- individual control of vanes 'simulates' row control by constraining angles of vanes within a stator row to have similar values thereby providing a direct comparison with IGC2

VANE CONTROL

- each vane is independently controlled by agent and vane angles may differ within a stator row as determined by control conditions

STOP - disables agents

OPTIMISE - enables / disables agent routines for maximising overall efficiency

FLOW VARIATION

- enables / disables random variation in flow angle at inlet to each vane

EXIT - closes IGC3 program

b) operating point panel

- displays current operating point data and also indicates the location of the row with stall margin less than zero

c) graphics window

- displays overall machine characteristics and stage characteristics as selected by user through updown control button

d) MAS window

- displays array of vane agents, the internal state of each being indicated by the same colour code as used in IGC2 and defined in the previous chapter.

e) user controls

- set-point and throttle setting trackbars
- up-down control for manually setting limiting value for overall stall margin

f) computed data tables

- user may display the following tables using the up-down control button:

stage data

	psi	Pex	phi	eff	R	rho	a0	a1	a2	a3	a4	smr	sms
IGV		0.977	0.816			1.225				0.000	13.104		1.000
1	1.024	1.617	0.837	0.902	0.500	1.194	13.104	43.880	13.112	43.876	13.110	1.000	1.000
2	1.024	2.349	0.837	0.902	0.500	1.707	13.110	43.883	13.113	43.881	13.112	1.000	1.000
3	1.024	3.161	0.837	0.902	0.500	2.225	13.112	43.900	13.118	43.897	13.117	1.000	1.000
4	1.024	4.042	0.837	0.902	0.500	2.746	13.117	43.878	13.113	43.881	13.114	1.000	1.000
5	1.024	4.988	0.837	0.902	0.500	3.269	13.114	43.874	13.113	43.875	13.113	1.000	1.000

STATOR VANE STAGGER ANGLE (Degrees) AT OPERATING POINT

IGV	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6				
1	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6				
2	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6			
3	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6		
4	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	
5	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6

DEVIATION FROM MEAN INLET FLOW ANGLE FOR STATOR ROW (Degrees)

IGV	0.8	-1.6	-1.8	1.5	1.8	-0.6	-0.2	-1.8	1.8	-0.8	1.6	1.8	-1.5	-1.8	0.6	0.2				
1	2.8	-1.0	0.8	1.3	1.5	0.0	-3.6	1.6	-1.3	-1.5	0.0	3.8	-1.6	-2.8	1.0	-0.8				
2	3.9	3.3	-0.3	-3.7	1.3	-3.4	1.1	-4.2	-3.9	-3.3	0.3	3.7	-1.3	3.4	-1.1	0.0	4.2			
3	-3.9	-0.9	-1.4	2.8	2.6	-0.5	-3.4	3.0	-1.3	3.4	-3.0	1.3	3.9	0.9	1.4	-2.8	-2.6	0.5		
4	-3.3	0.9	-3.3	2.3	-2.4	1.4	4.1	-3.1	-1.3	0.0	3.3	-0.9	3.3	-2.3	2.4	-1.4	-4.1	3.1	1.3	
5	-2.6	3.8	1.3	-2.4	-1.5	-0.9	-0.2	0.9	-3.5	-3.4	3.4	2.6	-3.8	-1.3	2.4	1.5	0.9	0.2	-0.9	3.5

STATOR VANE STALL MARGIN

IGV	0.299	0.320	0.116	0.103	0.260	0.327	0.031	0.296	0.259	0.272	0.114	0.047	0.343	0.078	0.075	0.054				
1	0.845	0.740	0.212	0.757	0.183	0.296	0.409	0.773	0.648	0.103	0.677	0.564	0.451	0.086	0.015	0.120				
2	0.358	0.572	0.358	0.585	0.271	0.580	0.007	0.486	0.424	0.197	0.510	0.201	0.774	0.391	0.424	0.209	0.296			
3	0.293	0.583	0.394	0.361	0.071	0.671	0.377	0.655	0.212	0.398	0.108	0.297	0.329	0.619	0.019	0.313	0.035	0.478		
4	0.548	0.670	0.434	0.800	0.722	0.635	0.194	0.862	0.777	0.411	0.273	0.151	0.387	0.021	0.100	0.187	0.627	0.159	0.045	
5	0.596	0.747	0.762	0.330	0.465	0.207	0.218	0.222	0.134	0.600	0.037	0.022	0.454	0.319	0.577	0.565	0.562	0.650	0.184	0.188

vane stall margin values < 0.025 highlighted by program

8.5.2 Flow Model (Module: *Runflow*)

Several of the functions, which make up this module, were modified to accommodate the introduction of the flow variation and the increase in number of agents. For the latter, the changes were mainly extensions of data arrays and the inclusion of additional 'sensor update' event flags necessary for thread synchronisation. The flow variation was implemented in two parts.

Firstly, a random coefficient in the range -1 to +1 is generated for each stator vane at the start up of the program. with the constraint that the algebraic sum of the coefficients for each stator row is zero. The coefficients are held in a 2-dimension array rc[iI][j]. The code for performing this task is shown in the panel below.

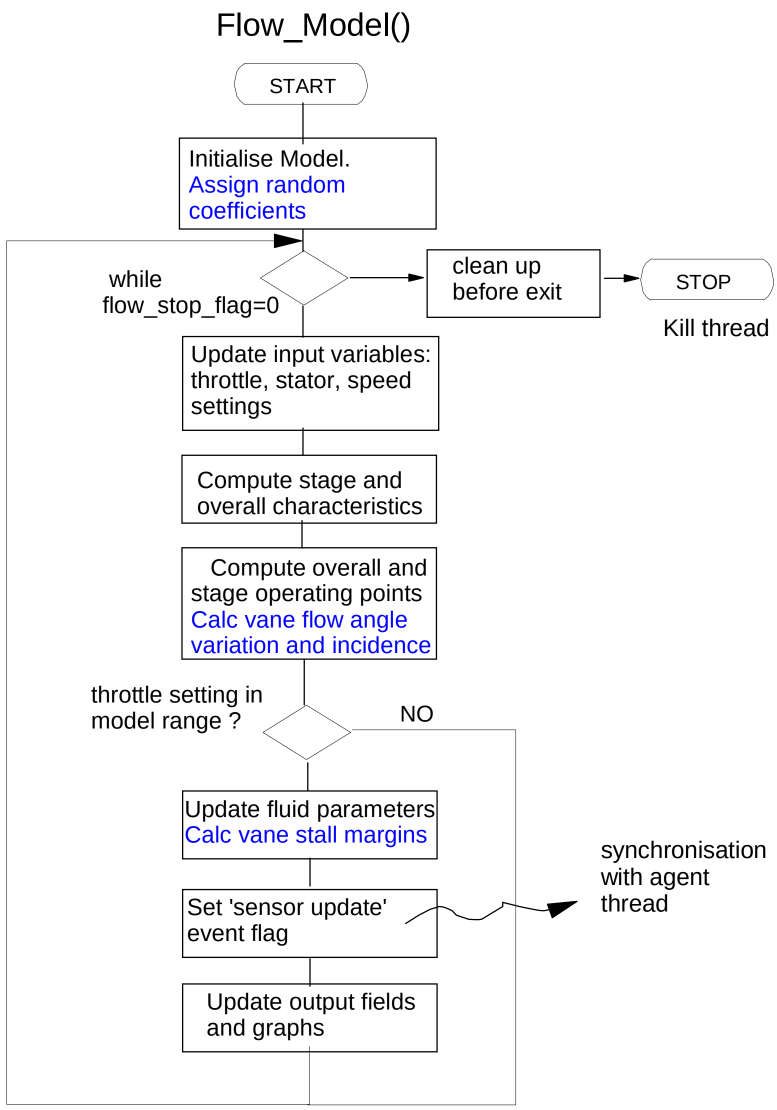
```
void assign_rc(int i)
/*  This routine assigns random values to coefficients for the vanes in stator  row, i, such that the algebraic sum of
the  coefficients  is zero. The process is basically to assign random values for the vanes in one half of the row   and
then  assign equal but opposite  (in sign) values for the vanes in the other half starting at a randomly selected vane in
that half. If the stator has an odd number  of vanes then the coefficent for  the 'odd' vane (which is also the starting
vane for the second half assignments) is set to zero. */

{
int mid,k,kx,j,jx;

  // determine number of vanes in half of stator row
  mid=floor(m[i]/2);  // doesn't matter whether odd or even number of vanes
  // assign random value in range +/- 1 to each vane in the 'lower' half of the row i.e. from vane 1 to 'middle' vane
  // inclusive
  for(j=1;j<=mid;j++) rc[i][j]=0.01*(random (200)-100);
  // select vane at random from 'upper' half of row to be 'odd' vane
  kx=mid+random(m[i]-mid);
  // starting with the vane after the 'odd' vane assign value to each vane in the upper half of row, equal to but of
  // opposite sign to the corresponding  value for the coefficient in lower half
  j=0;
  for(k=kx+1;k<=mid+kx;k++){
    j=j+1;
    jx=k;
    if(k>2*mid)jx=k-mid;
    rc[i][jx]=-rc[i][j];
    }
  // if odd number of vanes in row then reassign 'odd' vane value to last vane and assign 0 value to 'odd' vane - we
  // know rows 2 and 4 have odd number of vanes
  if((i==2)||(i==4)){
            rc[i][m[i]]=rc[i][kx];
    rc[i][kx]=0;
    }
}
```

The second part of the flow variation computation is carried out on every cycle of the flow model after the overall and stage operating points have been determined. It involves firstly calculating the inlet angle variation for each stator vane using the random coefficients and then the vane incidence. The calculation is only done if a change in overall operating point has been detected. The values of vane incidence are subsequently used to determine values of vane stall margin.

These modifications to the flow model are highlighted in the flowchart below.

Fig 8.6 Flow Model

Showing modifications for 'flow variation'

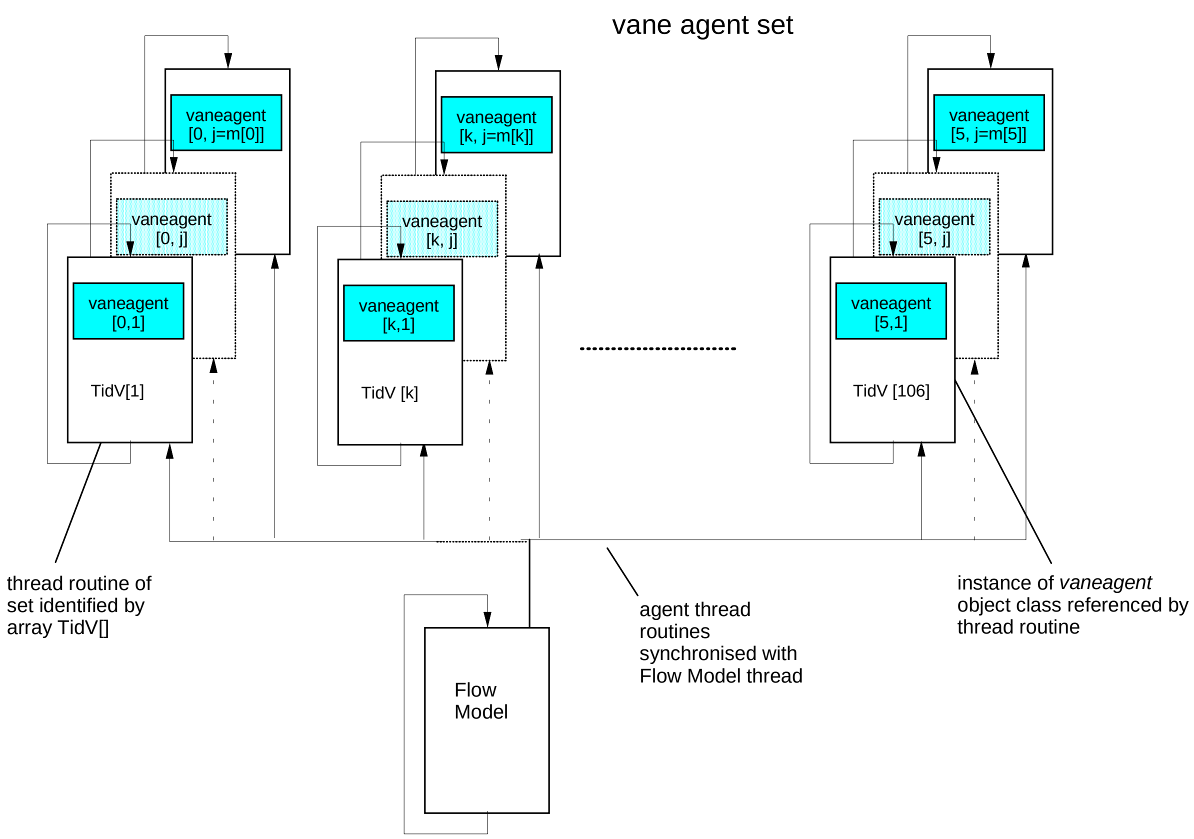


8.5.3 MAS Implementation

The vane agents were implemented in exactly the same manner as the row agents of the previous MAS. In this case, 106 program threads are created and the objects associated with vane agents are declared as a 2-dimensional array of the object class *vaneagent.*

The overall implementation is represented in the diagram below.

Fig 8.7 Overall Implementation of MAS



The implementation of the vane agent in terms of control structure, message passing and methods is generally very similar to that for row agents and much of the earlier source code was re-used. The main differences are briefly described below.

Firstly, the value of stator stall margin used by the vane agent is that generated by the flow model for the related stator vane rather than the mean of the stator row. However, if the user selects "row control" from the main menu then the vane agent will apply the mean stator row stall margin. This facility enables the vane agent system to "simulate" the behaviour of the row agents which is useful for comparison purposes.

To implement the alternative optimisation method required the introduction of a variable for the limiting stall margin value. This value is used in the condition statements, which invoke the methods for corrective action as indicated in the flow chart of Fig 8.8. The optimisation method is only active if this option has been selected on the main menu. If not selected, then the user can manually set the value for limiting stall margin using the control described in section 8.5.1.

When selected, the optimisation method is invoked if the following conditions are met:

a) pressure error (relative to set point) is within pre-set tolerance
b) corrective action is complete
c) current value of overall stall margin is less than 95%

The method simply involves the incremental increase of the limiting value of stall margin. This may be done in various ways but was implemented initially by means of the following statement:

```
lims=smo*(1+0.01);
```

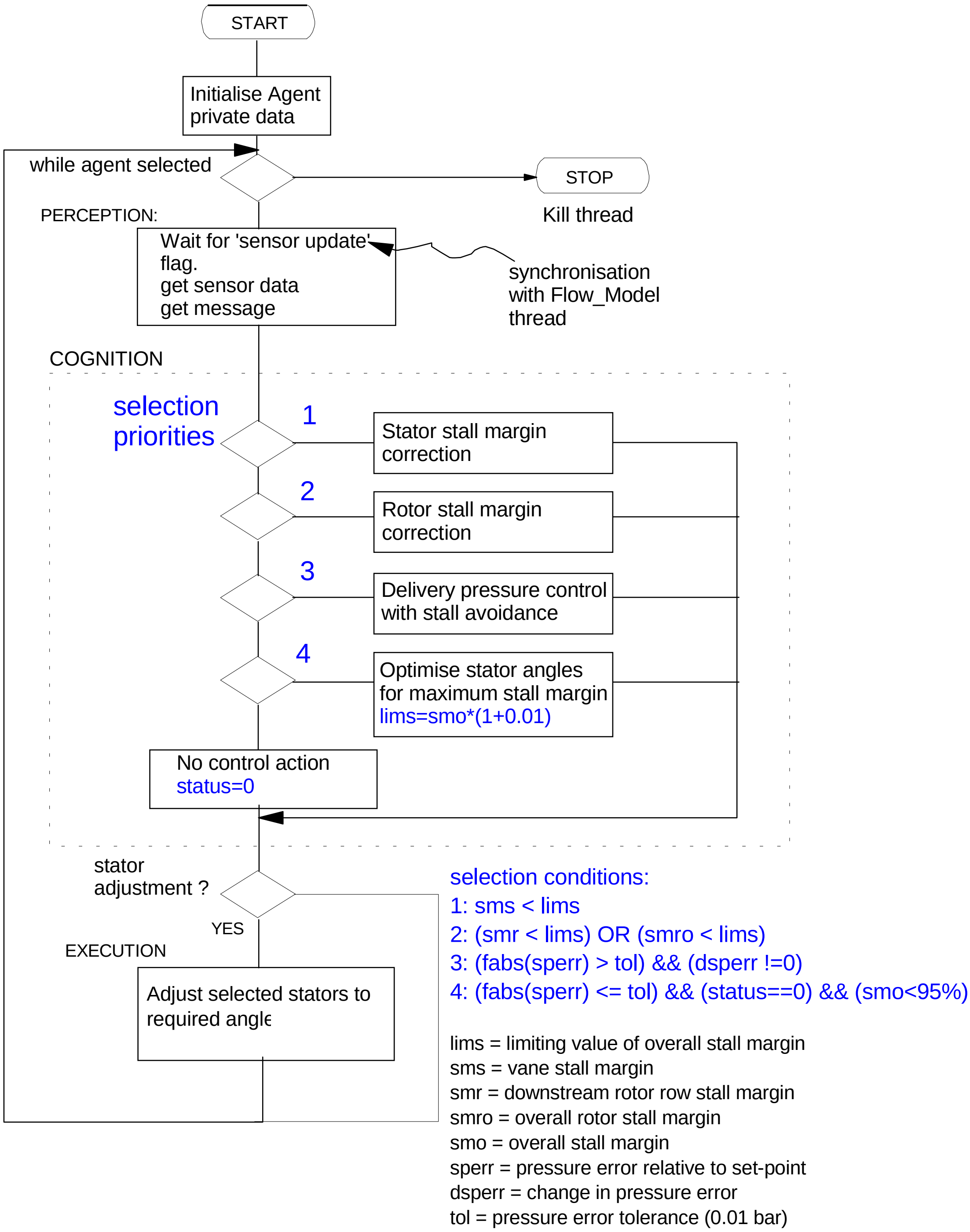
where: lims = limiting value for overall stall margin

smo = current value of overall stall margin

In this way, each time a stable operating point is reached within the tolerance of the set-point the stall margin limit is increased by 1% and one or more vane agents re-apply corrective action while others operate simultaneously to maintain the set-point. The process continues until the point when agents are unable to hold the pressure within the set-point tolerance.

Fig 8.8 Vane Agent Control Structure

Similar to Row Agent (Fig 7.12) - main differences highlighted



8.6 Simulation Trials

8.6.1 Validation of MAS Implementation

For this purpose, the flow variation feature was not selected. Firstly, the operating boundary of the compressor was obtained with the agents operating in 'ROW CONTROL' mode. This was repeated in 'VANE CONTROL' and the results compared.

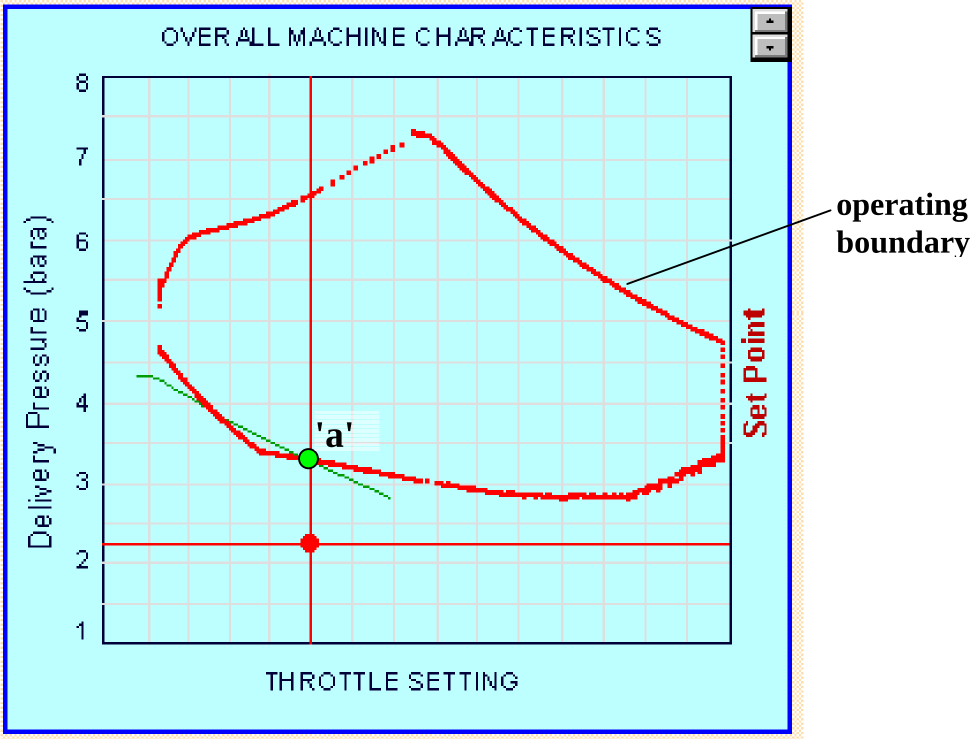
In 'ROW CONTROL', all vane agents are active but use the mean stator row stall margin as a control variable rather than that for the individual vane. This means that the vane agents associated with a given row experience the same values of input variables and therefore generate the same output value of vane angle. The result should therefore be identical to that achieved with the row agent system of IGC2. The operating boundary was obtained by the method described previously of adjusting the user controls so as to move the set point around the extremes of the pressure-throttle field. A screen image of the graphs window showing the operating boundary obtained is shown in Fig 8.9 overleaf. Also shown are data tables relating to a sample point on the boundary.

The operating boundary under row control appears to be identical to that obtained with the row agents in the simulation program IGC2. The data tables confirm that flow variation is not active and that at the particular sample boundary point the stall margins for all stator vanes approach zero. Also the data table for vane stagger angle shows that all angles are the same in a given row. (In practice, after traversing the boundary and returning to the sample point, minor differences, typically 0.1 degree, were noted.)

The corresponding results obtained when 'VANE CONTROL' was selected are shown in Fig 8.10. With no flow variation, an individual vane stall margin should be the same as the mean of the stator row to which it belongs so, again, all vane agents associated with a particular stator row should behave identically. The results show this to be the case. Since the operating boundary is the same as that obtained with 'ROW CONTROL' then it may be concluded that the MAS implementation is correct.

Fig 8.9 Operating Boundary with Row Control

Flow Variation Inactive



<u>Data tables corresponding to sample point 'a' on boundary</u>

vane angles same in a row

STATOR VANE STAGGER ANGLE (Degrees) AT OPERATING POINT

IGV	17.9	17.9	17.9	17.9	17.9	17.9	17.9	17.9	17.9	17.9	17.9	17.9	17.9	17.9	17.9	17.9				
1	31.1	31.1	31.1	31.1	31.1	31.1	31.1	31.1	31.1	31.1	31.1	31.1	31.1	31.1	31.1	31.1				
2	30.2	30.2	30.2	30.2	30.2	30.2	30.1	30.2	30.2	30.2	30.2	30.2	30.2	30.2	30.2	30.2	30.2			
3	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9		
4	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	
5	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	23.8	24.0	24.0	24.0	24.0	24.0	24.0	24.0

flow variation inactive

DEVIATION FROM MEAN INLET FLOW ANGLE FOR STATOR ROW (Degrees)

IGV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0				
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0				
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

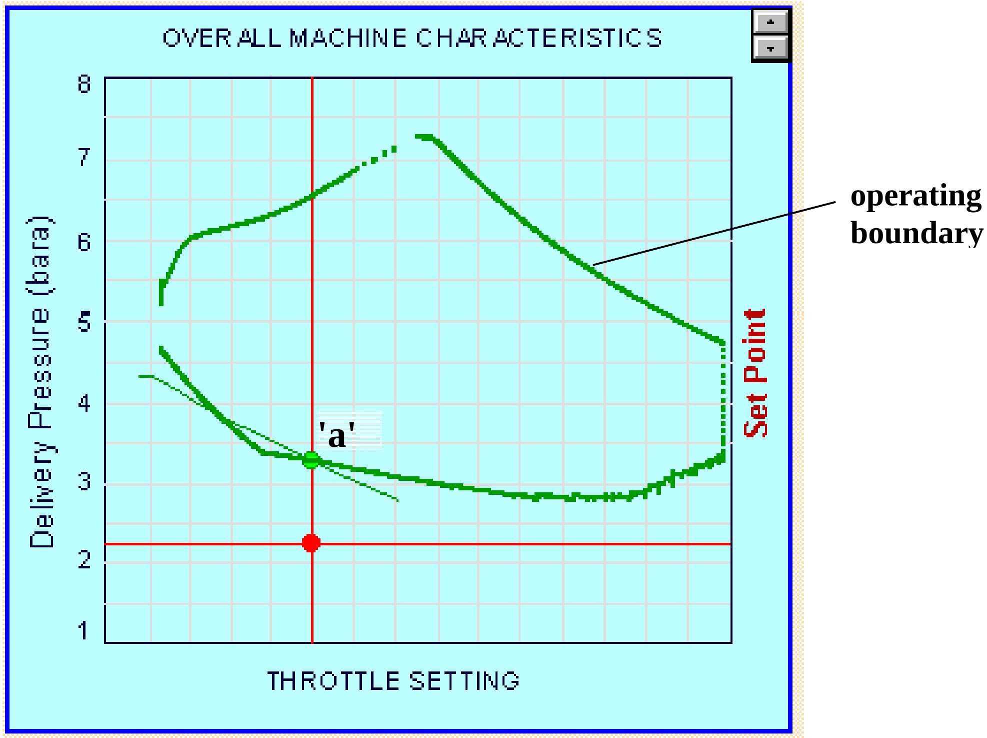
all stator vanes at 'zero' stall margin

STATOR VANE STALL MARGIN

IGV	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001				
1	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001				
2	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001			
3	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001		
4	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	
5	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001

Fig 8.10 Operating Boundary with Vane Control

Flow Variation Inactive



<u>Data tables corresponding to sample point 'a' on boundary</u>

vane angles same in a row

STATOR VANE STAGGER ANGLE (Degrees) AT OPERATING POINT

IGV	17.9	17.9	17.9	17.9	17.9	17.9	17.9	17.9	17.9	17.9	17.9	17.9	17.9	17.9	17.9	17.9				
1	31.1	31.1	31.1	31.1	31.1	31.1	31.1	31.1	31.1	31.1	31.1	31.1	31.1	31.1	31.1	31.1				
2	30.2	30.2	30.2	30.2	30.2	30.2	30.2	30.2	30.2	30.2	30.2	30.2	30.2	30.2	30.2	30.2	30.2			
3	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9	27.9		
4	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	25.8	
5	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0

flow variation inactive

DEVIATION FROM MEAN INLET FLOW ANGLE FOR STATOR ROW (Degrees)

IGV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0				
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0				
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

all stator vanes at 'zero' stall margin

STATOR VANE STALL MARGIN

IGV	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001				
1	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001				
2	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001			
3	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001		
4	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	
5	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001

8.6.2 Introduction of Flow Variation

The next trial investigated the effects of flow variation on the operating boundary.

Firstly, the case of 'ROW CONTROL' was considered with the menu selection 'FLOW VARIATION' turned 'ON'. The resulting operating boundary is shown in Fig 8.11 superimposed over that obtained in the previous trial without flow variation. As before, data tables for a sample point on the boundary are also shown.

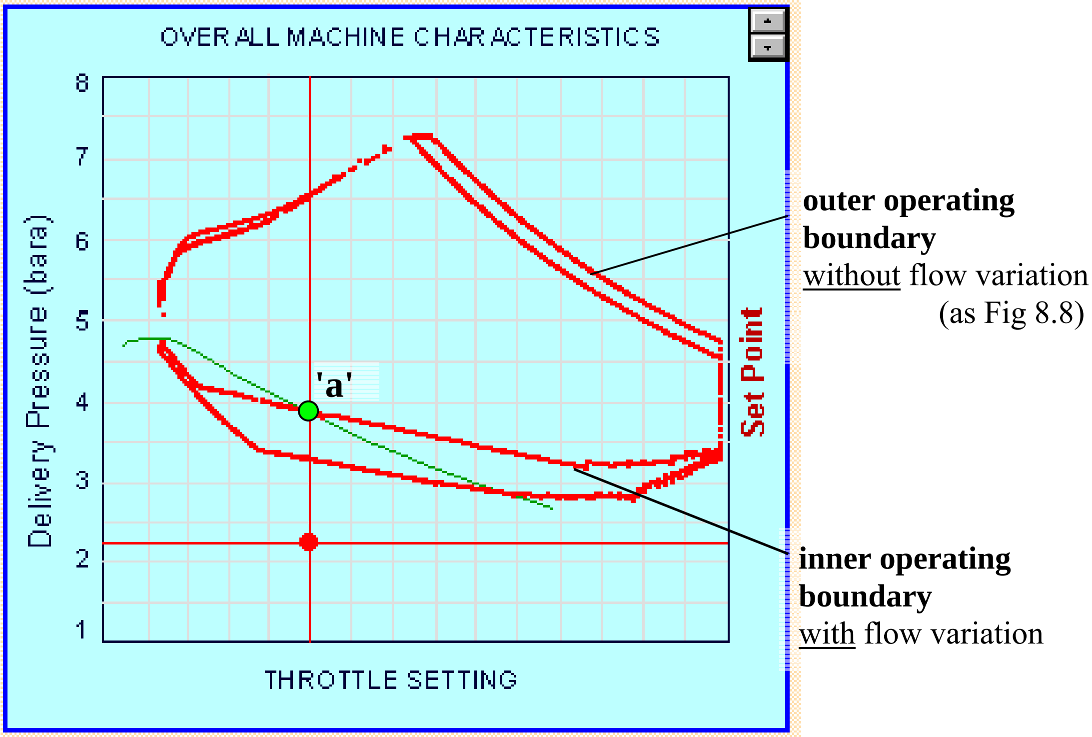
The results show that the effect of flow variation is to reduce the operating range, which is now limited by the minimum vane stall margin. The data tables show that vane stall margin varies throughout the stator rows as a result of the flow angle variation introduced. The vane angles are, necessarily, the same within a row. Only by having the ability to make individual adjustment of vane angles can this situation be controlled.

Repeating the trial with 'VANE CONTROL' selected enables vane agents to respond to local stall margin and thereby counteract the local variation in inlet flow angle. The results given in Fig 8.12 show that the original operating boundary, i.e. as obtained without flow variation, is fully recovered. In this case, the data tables show that the agents have adjusted vane angles to compensate for the flow angle variation and achieve a near-zero stall margin condition for all vanes in a row thereby maximising the operating range.

This trial demonstrated that if significant variation in operating conditions exist from vane to vane within a stator row then this would limit the effectiveness of a control system, which was constrained to adjust all vanes within a stator row equally. To deal effectively with such a situation requires independent stator vane control and the MAS of vane agents demonstrates this capability.

Fig 8.11 Operating Boundary with Row Control

Flow Variation Active



Data tables corresponding to sample point 'a' on boundary

vane angles same in a row

STATOR VANE ANGLE (Degrees) AT OPERATING POINT

IGV	16.1	16.1	16.1	16.1	16.1	16.1	16.1	16.1	16.1	16.1	16.1	16.1	16.1	16.1	16.1	16.1				
1	21.7	21.7	21.7	21.7	21.7	21.7	21.7	21.7	21.7	21.7	21.7	21.7	21.7	21.7	21.7	21.7				
2	20.4	20.4	20.4	20.4	20.4	20.4	20.4	20.4	20.4	20.4	20.4	20.4	20.4	20.4	20.4	20.4	20.4			
3	19.5	19.5	19.5	19.5	19.5	19.5	19.5	19.5	19.5	19.5	19.5	19.5	19.5	19.5	19.5	19.5	19.5	19.5		
4	18.2	18.2	18.2	18.2	18.2	18.2	18.2	18.2	18.2	18.2	18.2	18.2	18.2	18.2	18.2	18.2	18.2	18.2	18.2	
5	17.6	17.6	17.6	17.6	17.6	17.6	17.6	17.6	17.6	17.6	17.6	17.6	17.6	17.6	17.6	17.6	17.6	17.6	17.6	17.6

flow variation active

DEVIATION FROM MEAN INLET FLOW ANGLE FOR STATOR ROW (Degrees)

IGV	0.8	-1.6	-1.8	1.5	1.8	-0.6	-0.2	-1.8	1.8	-0.8	1.6	1.8	-1.5	-1.8	0.6	0.2				
1	3.1	-1.2	1.0	1.6	1.8	0.0	-4.3	1.9	-1.6	-1.8	0.0	4.3	-1.9	-3.1	1.2	-1.0				
2	4.4	3.7	-0.4	-4.3	1.4	-3.9	1.2	-4.8	-4.4	-3.7	0.4	4.3	-1.4	3.9	-1.2	0.0	4.8			
3	-4.3	-1.0	-1.6	3.1	2.8	-0.5	-3.7	3.3	-1.4	3.7	-3.3	1.4	4.3	1.0	1.6	-3.1	-2.8	0.5		
4	-3.6	0.9	-3.6	2.5	-2.6	1.6	4.5	-3.4	-1.4	0.0	3.6	-0.9	3.6	-2.5	2.6	-1.6	-4.5	3.4	1.4	
5	-2.8	4.1	1.4	-2.5	-1.6	-1.0	-0.2	1.0	-3.8	-3.7	3.7	2.8	-4.1	-1.4	2.5	1.6	1.0	0.2	-1.0	3.8

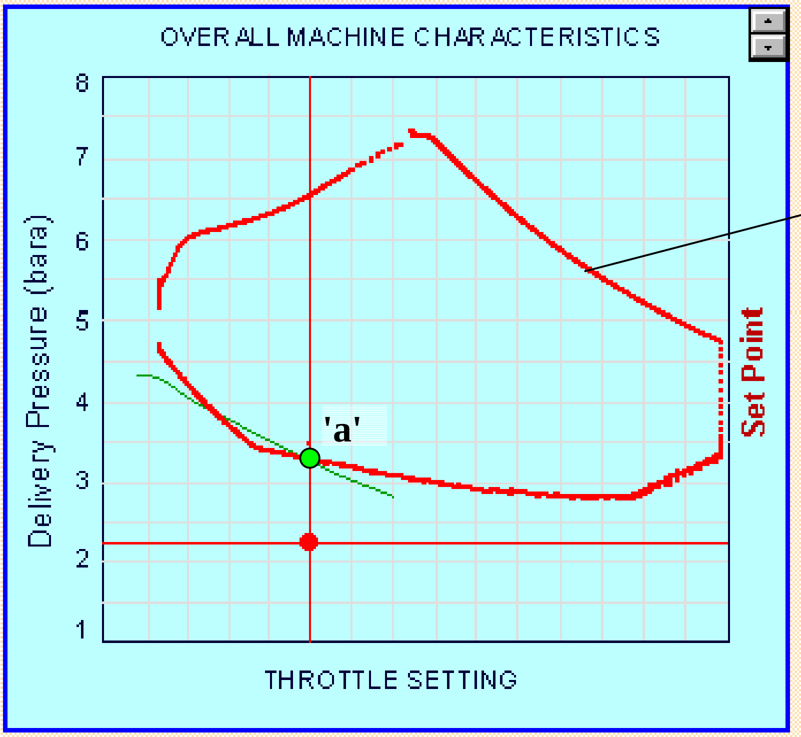
relatively few stator vanes at 'zero' stall margin

STATOR VANE STALL MARGIN

IGV	0.209	0.014	0.001	0.267	0.287	0.098	0.131	0.001	0.287	0.079	0.274	0.287	0.021	0.001	0.189	0.157				
1	0.606	0.255	0.433	0.479	0.496	0.352	0.002	0.509	0.226	0.209	0.352	0.703	0.196	0.099	0.450	0.272				
2	0.750	0.696	0.363	0.047	0.509	0.076	0.492	0.001	0.034	0.088	0.421	0.737	0.276	0.708	0.292	0.392	0.783			
3	0.002	0.269	0.224	0.604	0.584	0.309	0.050	0.625	0.236	0.657	0.083	0.471	0.706	0.439	0.483	0.103	0.123	0.398		
4	0.072	0.440	0.076	0.570	0.151	0.491	0.728	0.092	0.254	0.365	0.657	0.290	0.653	0.159	0.578	0.238	0.001	0.638	0.475	
5	0.106	0.688	0.451	0.129	0.207	0.257	0.319	0.412	0.029	0.036	0.633	0.563	0.002	0.219	0.540	0.463	0.412	0.350	0.257	0.641

Fig 8.12 Operating Boundary with Vane Control

Flow Variation Active



operating boundaries with and without flow variation coincide

Data tables corresponding to sample point 'a' on boundary

vane angles differ in a row

DEVIATION FROM MEAN INLET FLOW ANGLE FOR STATOR ROW (Degrees)

IGV	0.8	-1.6	-1.8	1.5	1.8	-0.6	-0.2	-1.8	1.8	-0.8	1.6	1.8	-1.5	-1.8	0.6	0.2				
1	3.4	-1.3	1.1	1.7	1.9	0.0	-4.7	2.1	-1.7	-1.9	0.0	4.7	-2.1	-3.4	1.3	-1.1				
2	4.8	4.1	-0.4	-4.7	1.6	-4.3	1.3	-5.3	-4.8	-4.1	0.4	4.7	-1.6	4.3	-1.3	0.0	5.3			
3	-4.7	-1.1	-1.7	3.3	3.1	-0.6	-4.0	3.6	-1.6	4.0	-3.6	1.6	4.7	1.1	1.7	-3.3	-3.1	0.6		
4	-3.8	1.0	-3.8	2.7	-2.8	1.7	4.8	-3.6	-1.4	0.0	3.8	-1.0	3.8	-2.7	2.8	-1.7	-4.8	3.6	1.4	
5	-2.9	4.3	1.5	-2.6	-1.6	-1.0	-0.2	1.0	-3.9	-3.8	3.8	2.9	-4.3	-1.5	2.6	1.6	1.0	0.2	-1.0	3.9

flow variation active

STATOR VANE ANGLE (Degrees) AT OPERATING POINT

IGV	18.7	16.3	16.1	19.4	19.7	17.3	17.7	16.1	19.7	17.1	19.5	19.7	16.4	16.1	18.5	18.1				
1	34.5	29.7	32.1	32.6	33.0	31.0	26.3	33.2	29.3	29.1	31.0	35.0	28.9	27.6	32.4	30.0				
2	35.0	34.3	29.8	25.5	31.8	25.9	31.5	24.9	25.4	26.1	30.6	34.8	28.6	34.5	28.8	30.2	35.0			
3	23.2	26.8	26.2	31.3	31.0	27.3	23.9	31.5	26.4	32.0	24.3	29.5	32.6	29.1	29.7	24.6	24.9	28.5		
4	21.9	26.8	22.0	28.5	23.0	27.4	30.5	22.2	24.3	25.8	29.6	24.8	29.6	23.1	28.6	24.1	21.0	29.3	27.2	
5	21.0	28.3	25.5	21.3	22.3	23.0	23.8	25.0	20.0	20.1	27.8	26.9	19.7	22.5	26.6	25.6	25.0	24.2	23.0	27.9

most stator vanes at 'zero' stall margin

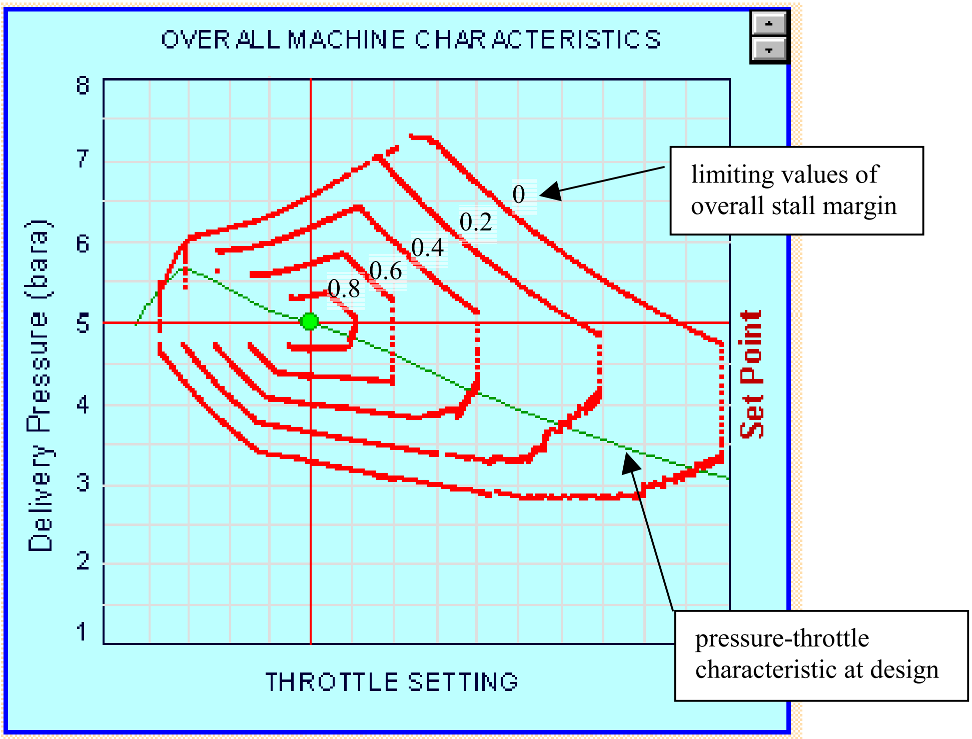
STATOR VANE STALL MARGIN

IGV	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000				
1	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.064	0.001	0.001	0.001	0.001				
2	0.003	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.040			
3	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.002	0.002	0.001	0.002	0.001	0.001	0.001		
4	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	
5	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001

8.6.3 Optimisation

Finally, simulation runs were carried out to investigate the effectiveness of optimisation of stator vane angles. Firstly, the underlying premise of the optimisation method was demonstrated by determining operating boundaries at different values of limiting stall margin. For this purpose, the *flow variation* and *optimise* selections in the main menu were turned 'OFF' and the limiting stall margin value was set manually by means of the user control. The results are shown in the screen image of Fig 8.13 which reveals a number of geometrically similar contours of differing size each representing the extent of operating range for a given stall margin limit.

Fig 8.13 Operating Boundaries at Various Stall Margin Limits

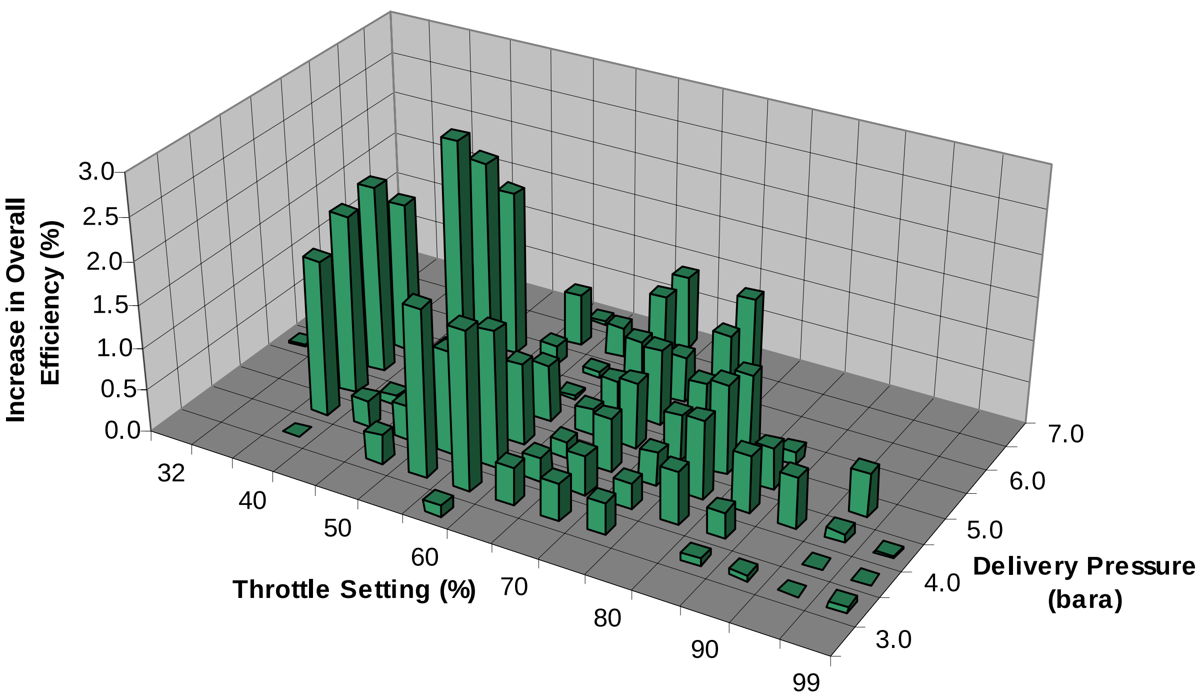


An operating point on a contour represents the closest the agents can achieve to a required operating point lying outside of the contour whilst not exceeding the stall margin limit. Thus, for that particular operating point, by definition, the overall stall margin is at a maximum.

When a change in operating point (set point or throttle change) is detected by the agents the stall margin limit is set to zero thus allowing the resulting operating point to lie anywhere within the larger boundary of Fig 8.13. However, as explained previously, the stator angles that obtain at the operating point as a result of agent action are not necessarily optimal since the agents are not designed to co-ordinate relative adjustment of stator elements during response to operating point change but only to ensure that local stall margins everywhere are not less than zero. In particular, it was noted previously that making small changes in set point from a point of zero stall margin often gave rise to sub-optimal stator settings.

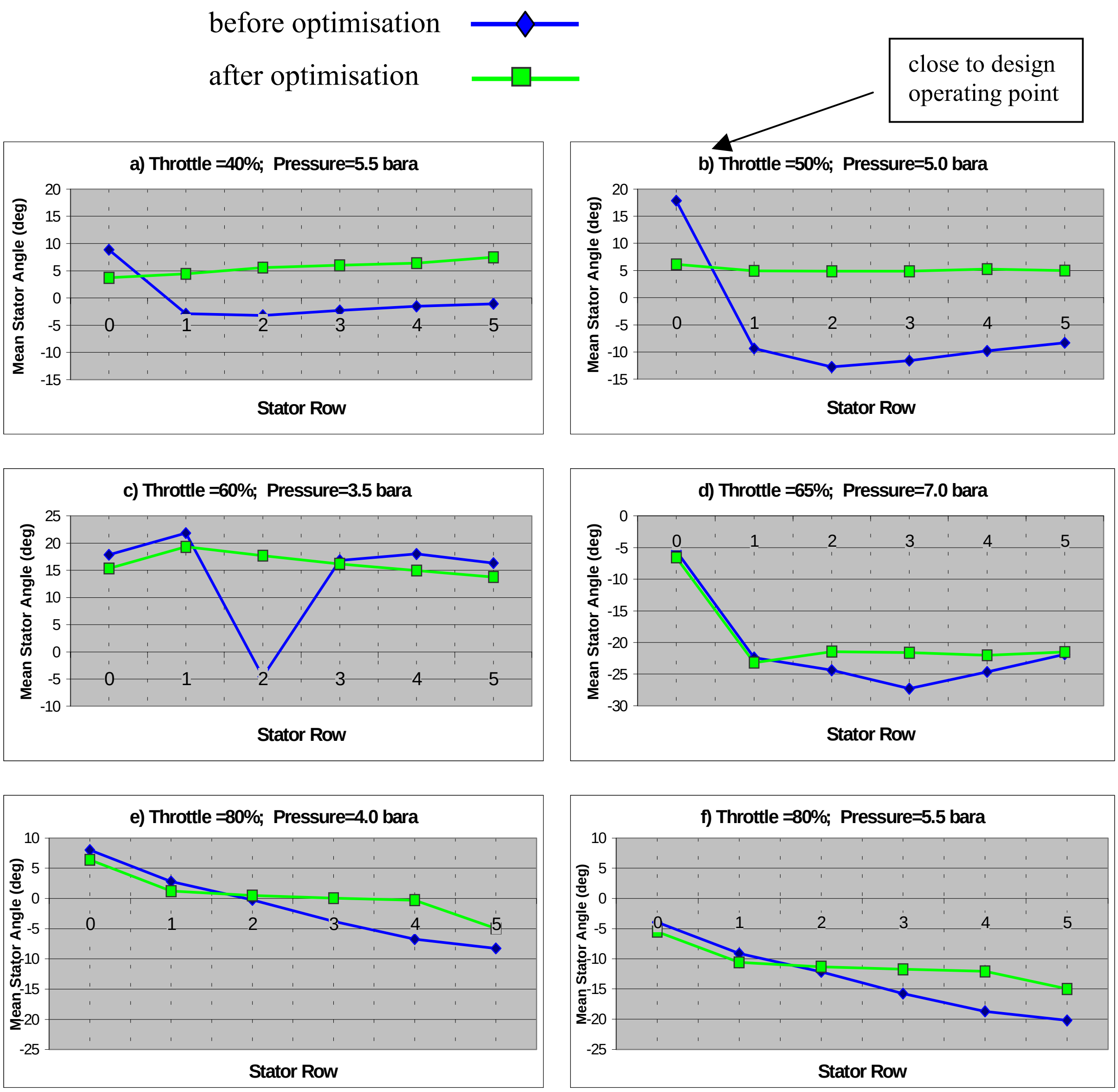
To investigate the improvement in efficiency that optimisation may produce, a similar trial to that described in the previous chapter for row agents was carried out. Two consecutive simulation runs were made; the first with the *optimise* selection 'OFF' and then with *optimise* 'ON'. Values of efficiency and mean stator angles at each of a large number (70+) of operating points over the pressure-throttle field were captured on each run. These are compared below.

Fig 8.14 Optimisation at Selected Operating Points



It was found that for most of the operating points an increase in efficiency occurred as a result of optimisation. The 3-D graph of Fig 8.14 indicates the location of operating points and the magnitude of increase in efficiency observed. At several points the increase was significant i.e. between 1% and 3%. The corresponding changes in mean stator angles for some of the operating points are shown in Fig 8.15 below.

Fig 8.15 Optimised Stator Angles at Selected Operating Points



As with the corresponding results obtained in the previous phase no significance is attributed to the location of operating points where increase in efficiency was noted. Were the trial to be repeated using a different sequence and with different magnitudes of operating point change then different results in terms of locations and changes in efficiency would be expected. Nevertheless, the overall effect of this alternative optimisation method is similar to that obtained in the previous phase and indicates that where there is scope to make improvement in the stator settings then the method is effective. Although the optimised results cannot be claimed to be 'best possible' it is noted that at the design operating point, Fig 8.15 b), the optimised result is very close to the design setting of equal angles.

The optimisation process was relatively quick, typically a few seconds, and the stator angle adjustments appeared generally to converge to their final values monotonically. This was in contrast to the method of phase 2, which, due to its sequential nature, took a much longer time to complete the optimisation process, and also involved many more adjustments of the stator rows.

The trials above were carried out without flow variation so that vane angles were always equal within rows. A brief trial was made with flow variation selected such that stall margins differed between vanes within rows at the start of the optimisation. The agents coped with this situation and resulted in stall margins for all vanes within a stator row being approximately equal although the time to complete optimisation was noticeably longer.

8.7 Conclusions

The results of this phase of the research lend support to the concept of an Intelligent Geometry Compressor in which all stator vanes are variable and independently controlled by autonomous agents. Computer simulation showed that such a system of 'vane' agents was able to achieve overall machine performance objectives within constraints set by local flow conditions experienced at the level of individual vanes.

The principal control algorithms employed by vane agents were direct adaptations of those used by row agents in the multi-agent system of Phase 2 and were found to be equally effective. The reservations remain, however, about the need to include specific pressure-flow criteria in the cognition algorithm used by the IGV agents (ref. Fig 7.35) and also about the need to provide all agents with overall performance information.

The results of the optimisation trials appeared to validate the approach of maximising stall margin as a means of increasing efficiency at a given operating point. The operation of the optimisation process concurrently by all agents gave advantages in speed and in reduced number of adjustments compared to the 'direct' method used by the MAS of phase 2.

CHAPTER 9

Physical Design of IGC

The body of the research presented in the preceding chapters has focused on the conceptual design of an intelligent geometry compressor (IGC) based on the paradigm of a multi-agent system. In this chapter, brief consideration is given to some of the implications arising for the physical design of such a machine.

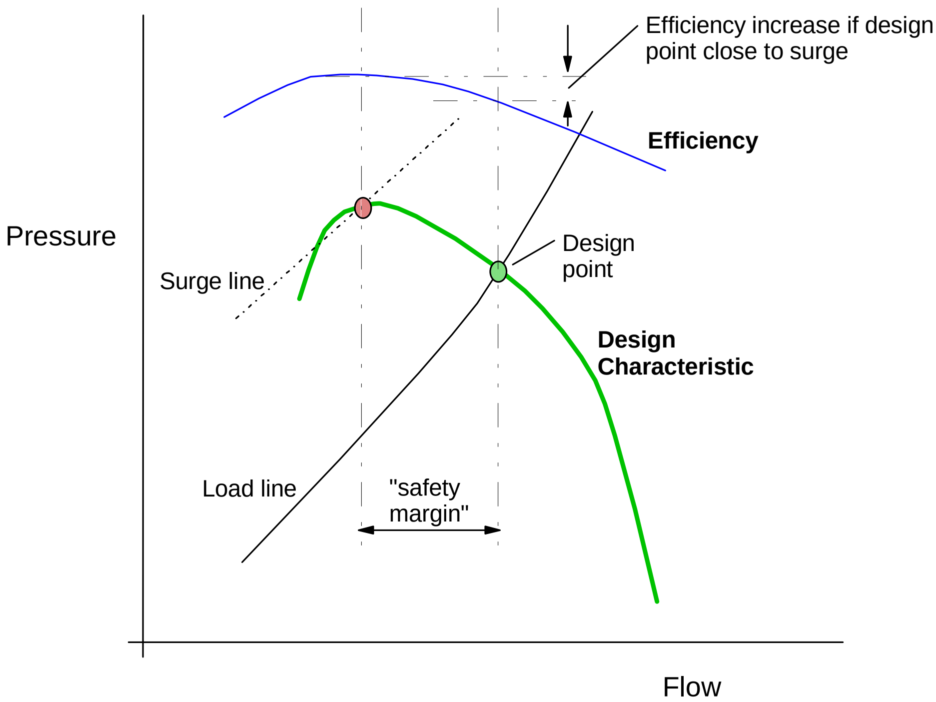
Physical Form of IGC

The underlying physics which govern the working principle of an axial flow compressor apply regardless of whether the machine has fixed or variable geometry. Thus the basic form of an IGC is bound to resemble that of a conventional machine in being a combination of compression stages each formed by a rotor-stator pair. However, the multi-agent system paradigm implies more than just a control system "bolted on" to an otherwise conventional design of compressor. There are two reasons for this statement. Firstly, the overall operating characteristics of the IGC, as borne out by the simulation results in this research, are very different from those of a fixed geometry compressor. Operationally, the IGC is a different machine. Secondly, the multi-agent system is conceived as a network of distributed agents each of which comprises software, hardware and mechanical elements and all of which need to be embodied effectively and economically in the structure of the machine. Both of these factors need to have a major influence on the physical design of the IGC from the outset if the end product is to fully meet the requirements of a particular application in the most effective and economic way.

Operating Envelope

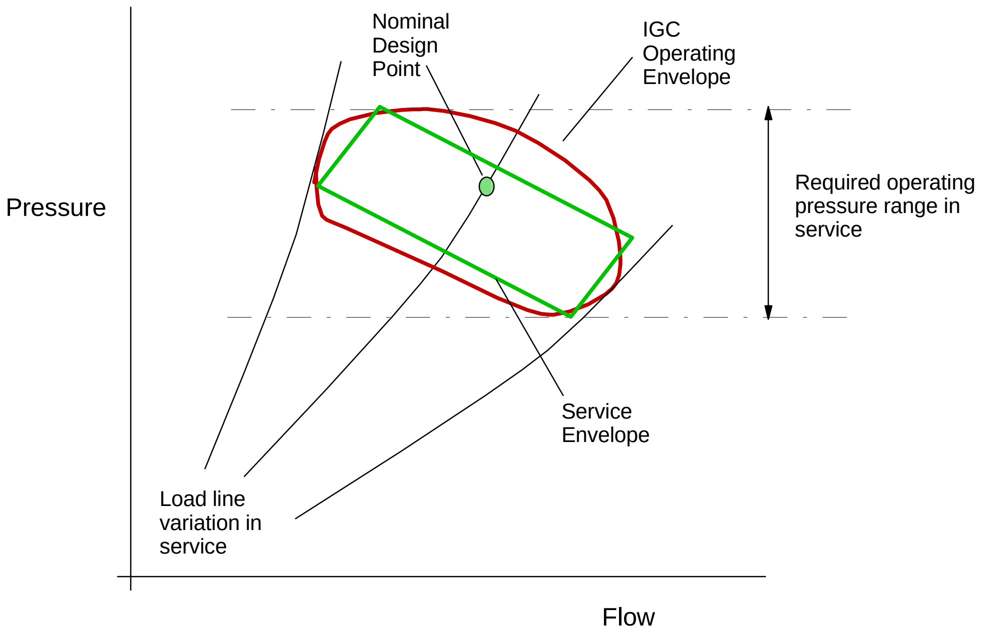
Typically, a conventional axial flow compressor is designed to achieve a particular operating point at design as set by the application requirements. The physical dimensions, number of stages and detail design of the blading are all determined such that the design operating point is achieved at the maximum value of overall efficiency (Gresh 1991). Provision for unplanned (but expected) variation in the operating conditions away from design is made by setting the actual design point away from the ideal thus providing a "safety margin" with respect to the surge line at the expense of efficiency. The sketch of Fig 9.1 serves as a reminder of this point. In the design of an IGC such provision is unnecessary and thus the physical design, especially of blading, may be optimised for a nominal design point much closer to the ideal and thus achieve improved efficiency in service.

Fig 9.1 Design Point for Fixed Geometry Compressor



The overall performance characteristic of the IGC is represented as a region of the delivery pressure-flow (or throttle setting) map i.e. ***operating envelope*** with the boundary defined by some limiting operating condition. In this research the limiting condition was defined in terms of static stall margin but alternative definitions may be applied. Where an application involves significant planned variation in pressure-flow requirements then these, ideally, need to be expressed in a way that defines a ***service envelope*** and a nominal design point. The latter may represent, for example, the point within the envelope where operation is most frequent. The physical design of the IGC then aims to produce an operating envelope for the machine which just matches the service envelope and maximises efficiency at the nominal design point as shown in Fig 9.2 below.

Fig 9.2 Operating Envelope for IGC



Solutions may be optimised for the number of compressor stages (overall dimensions and weight), stage loading (blade and vane material choice), the degree of variable geometry involved or some combination of these and other factors. One interesting and important point that needs to be considered in the design process is that whereas the physical unit on which the machine design is based for compression is the stage i.e. rotor + downstream stator, the physical unit for agent control is the stator + downstream rotor. This will be of significance when deciding the most suitable form and geometry of adjacent rows of blades and vanes to achieve the required operating envelope.

Degree of Variable Geometry

As mentioned, the degree of variable geometry included in the IGC may be open to consideration. The simulation results showed that variable stator rows contributed differently to the overall characteristics of the machine according to the relative position of the row and the particular operating point. Thus, depending on the service requirements the most effective solution may be a combination of variable and fixed geometry stator rows. The simulation results suggest that variable geometry for the IGV and other front end stator rows would always be most effective whereas the advantage may be marginal for the 'middle' rows. However, it is difficult to generalise and each application would need to be studied case by case in respect of the particular operating envelope being sought.

Feasibility

The economic and technical feasibility of agent implementation clearly has a crucial effect on the IGC design and the degree of variable geometry included. The MAS developed in the preceding chapters were based, respectively, on 'row agents' and 'vane agents' according to whether the vanes in stator rows were controlled collectively or individually. Since in either case all vanes must be movable, then a key issue is the choice of actuation method. For a row agent system a method such as that used by Riess and Blöcker (1987) could be used, in which a single actuator operates through an adjusting ring mechanism to cause the simultaneous and equal movement of each vane in a stator row (ref. Chapter 2 Fig 2.4). This method appears to be widely used within the field of turbocompresors. Alternatively, a motor may be applied to each vane independently in a similar arrangement to that used by Paduano et al (1993) for the IGV of his experimental machine (ref. Chapter 2 Fig 2.5). Individual vane actuation is necessary for realising a vane agent MAS.

The vane agent is conceived as a 'physical' agent and embodies all component parts, including the vane, necessary to achieve its functional objectives. This concept is most faithfully realised by considering the vane agent as a ***mechatronic*** unit and approaching the design in a multi-disciplined fashion as described in Chapter 3. In this way, the design and packaging of the electronic and mechanical parts for control, actuation and local sensing functions may be achieved most economically and combined with a stator vane as a self-contained sub-assembly. The embedded microcontroller in the vane agent would support communication with other agents using the standards and protocols such as CORBA and CAN described in chapter 3. The vane agents within a row are expected to be common units but may differ from row to row because of difference in vane size and related actuation requirements. Nevertheless, some standardisation of component parts would be possible.

The agent systems in this research have been designed to achieve steady state operating characteristics and the dynamics of the compressor system have not been considered. In particular, the need in practice to cope with rotating stall may lead to agent methods for 'active' stall control such as that investigated by Paduano (1993). In which case, the physical design of the IGC would require vane agents for the IGV row but other rows may be adequately managed by row agents. Thus hybrid agent systems, comprising both vane and row agents, may offer effective solutions in some applications should the cost of vane agents for all variable rows be prohibitive. In general, it would be expected that the dynamic response of vane agents would be superior to that of row agents simply from considerations of the relative mass of moving parts involved.

It has been proposed that agent physical design incorporate the sensors required for detecting local changes in the fluid environment. In addition, MAS operation requires information about the overall machine performance which is broadcast to all agents. Thus remote sensors for such parameters as pressure, flow and stall onset, are required and these may be 'smart' in that they process the sensed data and transmit directly on the agent network or, perhaps, a 'specialist agent' is introduced which collects the raw sensor data and handles processing and transmission.

Finally, in considering cost implications for the IGC, it might be noted that much of the cost involved would relate to the provision of variable geometry and sensors. This much would be the same whichever approach was adopted for system control given the same performance objectives. Cost associated with the physical distribution of the control hardware in agents would be expected to be a relatively small part of the total system cost.

CHAPTER 10

Conclusions

In this research a multi-agent systems (MAS) approach has been applied to the conceptual design of an Intelligent Geometry Compressor (IGC) in order to optimise performance and considerably enlarge the useful operating envelope of the machine. This has involved the development of a computer simulation program, which has supported the design task and which also demonstrates the enhanced performance that can be achieved by the IGC.

The hypothesis underlying this research is that the multi-agent system (MAS) paradigm offers an appropriate and effective basis for the design of intelligent machines, particularly in those cases where an alternative approach based on centralised control would prove difficult to apply. The effectiveness of the multi-agent system approach may be assessed in terms of the facility with which the design solution was achieved and implemented, and by an evaluation of the performance of the IGC as indicated by the results of computer simulation.

MAS Design

The MAS design approach, as with all approaches to the design of complex systems, begins with a decomposition of the design problem. For multi-agent systems the paramount objective is autonomy for the agents of the resulting system. This biases the decomposition towards physical criteria and leads naturally to the association of reactive agents with the manipulable parts of the machine structure. Thus for the IGC the design that emerged was based on a set of physically distributed reactive agents which are perceived as component parts of the machine. The agents are essentially identical and each agent is logically and physically associated with a particular variable geometry element. In the system design, developed in Phase 2 of the research, the variable element was chosen to be a stator row. In this case, the MAS included 6 reactive *row agents*, one for each of the variable rows of a hypothetical machine. The system was extended in Phase 3 of the research to accommodate individually variable vanes within stator rows such that the 6 row agents of the original system were replaced by 106 *vane agents*.

The identification of reactive agents based on assignment to chosen variable elements was a straightforward first step in the MAS design approach. The need to include other agents, or not, in the system design was not as clear-cut and several possible options were apparent. The MAS of Phase 2 eventually included an additional agent to facilitate the optimisation process. This decision was based on an assessment of the feasibility of the reactive agents to carry out the tasks necessary to fulfil the overall objectives of the IGC. The key factors influencing the assessment were:

1) The scale and difficulty of the tasks involved
2) the extent and complexity of co-operation required between reactive agents
3) the availability to agents of system-level information

These factors are application-specific and could only be considered with the knowledge of appropriate potential control strategies and methods. Such knowledge was largely derived from the analysis and simulation trials of Phase 1, which served to establish the principal relationships between the manipulated and control variables of the system.

The MAS concepts drawn from the published literature were helpful in providing a framework for the design solution. The simple *perception-cognition-execution* model was adopted for the internal structure of agents and a *subsumption-type architecture* was successfully used to organise the row agents' cognitive tasks. The interaction requirements of the MAS design led readily to an *agent network* system architecture within which agents communicated by means of a *message-passing* scheme. The level of agent co-operation required, however, was very limited. This was because the relative priorities of the IGC objectives were easily pre-determined and so conflicts could be resolved directly without the need for sophisticated negotiation between agents. Accordingly, the message-passing scheme was very simple and sufficient only to support the exchange of agent internal state information.

Thus the main elements of the MAS solutions developed in this research were arrived at by a rational process without too much difficulty. Far greater effort was expended in the detailed development of the control strategies and rules employed by the agents.

These were arrived at by a combination of engineering analysis and, mostly, trial and error using the computer simulation program. In this respect, simulation formed a crucial part of the research and design methodology. Firstly, it enabled proposed MAS design solutions to be demonstrated and evaluated. Secondly, the use of simulation provided valuable insight into system behaviour that inspired the successive cycles and phases of system development.

MAS Implementation

The implementation of the multi-agent system in software proved to be very straightforward. Each agent was implemented as an 'object + thread' in order to provide the characteristics of an autonomous entity. This representation was particularly efficient when dealing with a number of agents of the same type. Thus the row agents of Phase 2 and the vane agents of Phase 3 were easily created and run by means of simple array structures. The use of a common object class for a given agent set also minimised the amount of program code required since all objects in the same class access the same object methods. The message passing scheme could have been implemented by means of global variables purely for the sake of simulation. Instead, appropriate object methods were developed for this purpose which provided a more realistic representation of the sort of communication mechanism that might be employed in a real-world distributed system using a standard such as CORBA.

IGC Performance

The effectiveness of the multi-agent systems in achieving the IGC performance objectives is evident from the simulation results captured in Phase 2 and Phase 3. The potential for enhancing axial compressor performance through the control of variable internal geometry has been widely recognised for many years as the literature references revealed. But, beyond a few isolated experimental efforts such as that by Riess and Blöcker (1987), relatively little progress appears to have been made towards a control system which would enable this potential to be fully realised. The limitations of the compressor flow model used to evaluate the MAS means that caution needs to be exercised in drawing conclusions about real-world compressors in which flow is far from ideal and system behaviour is complicated by the dynamics of the machine and fluid. Nevertheless, the demonstration of effective automatic control over the steady state characteristics of variable geometry is a significant step.

The broad strategy employed in the IGC was in two parts. Firstly, agents react autonomously to changes in flow conditions in order to rapidly configure the internal geometry to provide a new stable operating point. Once steady state conditions have been recovered, the second part of the strategy is to perform an optimisation process to maximise overall machine efficiency at the new operating point.

An important feature of the first part of the strategy is that during a change in operating point there is no explicit mechanism for controlling the relative adjustment of the variable stators or vanes. The respective agents make such adjustment completely independently. However, each agent continually modifies its action to avoid local stall conditions and, since the effects of such action alter the flow conditions at neighbouring locations, in so doing it influences the actions of other agents. In addition, should a particular agent be unable to cope with local conditions then it will broadcast this situation and thereby further affect the actions of other agents. In this way, the net result of independent agent action, focused on local goals, is to propagate a sort of 'incidental' co-ordination of action throughout the agent set which is beneficial to overall machine performance. This behaviour reveals the essence of a reactive multi-agent system and, albeit in a small way, provides a glimpse of what may be regarded as 'emergent' intelligent behaviour. The idea that intelligent behaviour emerges from the interaction of various simpler behaviours is a recurring theme in multi-agent research, and one that is particularly associated with reactive architectures (Weiss 1999).

The MAS operation described above appears to explain why, contrary to expectations, the stator settings following an operating point change often retain a near-optimal relationship. Even so, the simulation trials revealed instances where this was not the case and, on such occasions, the optimisation strategy proved effective in maximising the overall efficiency at the particular operating point. In the Phase 2 MAS, the optimisation process was controlled by an 'efficiency' agent which effectively scheduled the sequential action of row agents. In this particular case, the optimisation strategy was fairly simplistic and could have been implemented by the row agents alone. However, the inclusion of a specialist agent facilitates the implementation of other, more sophisticated, optimisation strategies, perhaps based on intelligent techniques, which might be considered in future.

The MAS of Phase 3 employed the same general control strategy as described above and virtually identical code was used to implement the vane agents. The approach to optimisation, however, was different to that used in Phase 2 and did not require the provision of a specialist agent. The main achievement of Phase 3 was to demonstrate the potential of a system comprising a relatively large number of autonomous agents. The underlying premise of the system is that the stall condition at individual stator vanes governs the stall condition of the local row and hence, ultimately, constrains the performance of the whole machine. The introduction of independently variable stator vanes and associated agents into the MAS allows conditions at the locality of each vane to be controlled. To enable the Phase 3 MAS to be evaluated, the compressor flow model was extended to admit the occurrence of flow variation at each stator vane. The simulation results showed that the vane agent system successfully compensated for such variation and enabled the full operating range of the IGC to be maintained. The behaviour of vane agents, not surprisingly, was the same as observed for the row agent system of Phase 2 and exhibited the same 'incidental' co-ordination of vane adjustment during changes in operating point.

The optimisation strategy of the Phase 3 MAS was based on the concept of maximising stall margin at a given operating point as an indirect way of maximising efficiency. This proved to be equally as effective as the strategy employed in Phase 2 with the added benefit of being a concurrent, as opposed to sequential, process and therefore was significantly faster. Of interest, from the point of view of software implementation, is the observation that there was no noticeable difference in speed of operation between the 106-agent system of Phase 3 and the 7-agent system of Phase 2.

General Conclusion

The experience of the research supported by the specific results and observations from many simulation trials, as summarised and explained in the foregoing narrative, leads to the general conclusion that multi-agent systems can provide an effective and novel approach to the design of an intelligent geometry compressor. By implication, this conclusion may be extended to other intelligent machine applications where similar opportunity to apply a distributed control solution exists. Thus the hypothesis of the research is supported.

The general conclusion above should be qualified by a reminder of the main limitations of the work undertaken in this research. First must be the assumptions made relating to the compressor flow model and in particular to stall. In this work, stall has been treated as a property of a blade or row and assumed to be a continuous variable amenable to measurement so that the occurrence of stall conditions may be anticipated. In reality, the onset of stall is unlikely to be so conveniently determined and this may have profound effects on the practical feasibility of the control methods demonstrated here. The absence of dynamic flow phenomena from the compressor flow model and the assumption of universally available system-level information means that this work is unable to say anything about control system stability. In an application such as the MAS of Phase 3 involving over 100 concurrent control entities and all having some degree of influence on overall system control variables there must be a concern about stability of operation under all circumstances.

A further general conclusion is drawn from the review of related areas of technology reported in Chapter 3. This revealed a strong synergy between MAS and Mechatronics such that the two fields, together, could form an effective overall strategy for the design and realisation of intelligent machines. Within such a strategy there is scope to apply methods and techniques from the field of Intelligent Control to the design of cognitive agents.

Suggestions for Further Work

Finally, the scope for further work arising from this research is considerable. Even with the limitations described above the existing system designs and simulation programs can be used to pursue two particular areas of work. Firstly, there is need to improve the generality and robustness of some of the control methods and related computer code. The situation when the IGV and the immediate downstream rotor row approach stall conditions is a case in point as is MAS operation, generally, at points on the boundary of the operating range. The investigation of optimisation by means of 'intelligent' techniques in the efficiency agent of Phase 2 might also be worth consideration. The second area of work arises from the discussion of chapter 9 concerning the physical design of the IGC. Of interest is a systematic study of different machine configurations (i.e. numbers of stages), physical proportions and different blade geometries in order to reveal the novel effects that MAS may have on machine design for given specific applications, mindful of the mechatronic approach advocated.

From the comments made earlier, it is desirable to extend the current work to encompass dynamic system behaviour and to study the stability of multi-agent systems. For this purpose, the simulation program needs to be re-developed to include an appropriate dynamic flow model. At some point, it would be interesting to pursue the development of a real IGC by linking the MAS software with real variable geometry compressors in the laboratory.

REFERENCES

ALAN, R. 2001. *Local Wireless Communication: Industrial Applications.* ERA Report 2001-0364. ERA Technology Ltd. ISBN: 0 7008 0742 X

ANSAKLIS, P.J. and PASSINO, K.M. 1993. *An Introduction to Intelligent and Autonomous Control.* Kluwer Academic Publishers.

BARBUCEANU, M. and FOX, M. 1997. Integrating Communicative Action, Conversations and Decision Theory to Coordinate Agents. In *Proceedings of Autonomous Agents'97*, Marina del Rey, CA.

BLOCH, H. P. 1996 *A Practical Guide to Compressor Technology*. McGraw-Hill.

BRADLEY, D.A., DAWSON, D., BURD, N.C., and LOADER, A.J. 1991**.** *Mechatronics - Electronics in products and processes*. Chapman & Hall.

BROOKS, R.A. 1986. A Robust Layered Control System for a Mobile Robot. In *IEEE Journal of Robotics and Automation,* 2(1):14-23, 1986.

BROOKS, R.A. 1990. Elephants Don't Play Chess. In *Robotics and Autonomous Systems*, 6.

CALVIN, W.H. 1996. *How Brains Think.* Science Masters Series. London: Weidenfeld & Nicolson.

CAMP, T. R. and HORLOCK, J. H. 1993. *An Analytical Model of Axial Compressor Off-Design Performance.* International Gas Turbine and Aeroengine Congress and Exposition. Cincinnati, Ohio, May 24-27.

CUMPSTY, N. A. 1989. *Compressor Aerodynamics*. Longman Scientific & Technical.

ENGLEMORE, R. and MORGAN, T. 1988. *Blackboard Systems,* Addison-Wesley.

ERMAN, L., HAYES-ROTH, F., LESSER, V. and REDDY, D. 1980. *The HEARSAY II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty.* ACM Computing Surveys. 12, 1980.

ESCURET, J. F. and ELDER, R. L. 1993. Active Control of Surge in Multi-Stage Axial-Flow Compressors. In*: Proceedings of the Institution of Mechanical Engineers Part G Journal of Aerospace Engineering,* V 209 N G4 1995 pp249-258.

FERBER, J. 1999. *Multi-Agent Systems*. Addison-Wesley.

FININ, T., FRITZON, R., McKAY, D. and McENTIRE, R. 1993. *KQML - A Language and Protocol for Knowledge and Information Exchange.* Tech. Report, University of Maryland, Baltimore.

GRESH, T. M. 1991. *Compressor Performance*. Boston and London: Butterworth-Heinemann.

GUIDA, G. and TASSO, C. 1994. *Design and Development of Knowledge-Based Systems*. John Wiley & Sons.

HORLOCK, J. H. 1958. *Axial Flow Compressors.* Butterworths Scientific Publications.

HOSNY, W. M., LEVENTHAL, L. and STEEKEN, W. G. 1991. *Active Stabilization of Multistage Axial-Compressor Aerodynamic System Instabilities*. The American Society of Mechanical Engineers, New York. Paper 91-GT-403.

HOWELL, A. R. 1945. Fluid Dynamics of Axial Compressors. In: *Proceedings of I.Mech.E.*, volume 153, page 441. London.

HOWELL, A. R. and BONHAM, R. P. 1950. Overall and Stage Characteristics of Axial-flow Compressors. In: *Proceedings of I.Mech.E.*, volume 163, page 235. London.

HOWELL, A.R. and CALVERT, W. J. 1978. *A New Stage Stacking Technique For Axial-Flow Compressor Performance Prediction*. Gas Turbine Conference, London, April 9 - 13, of the American Society of Mechanical Engineers. Paper No. 78-GT-139.

JANG, J-S. R. and SUN, C-T. 1995. Neuro-Fuzzy Modelling and Control. In: *Proceedings of IEEE,* Volume 83, No 3, pp 378-406.

JENNINGS, N. R. 1993. Commitments and Conventions: The Foundations of Coordination in Multi-Agent Systems.
In: *The Knowledge Engineering Review,* 2(3): pp223-250.

JENNINGS, N. R. 1994. *Co-operation in Industrial Multi-Agent Systems.*
Volume 43, World Scientific Press.

JENNINGS, N. R. and WOOLDRIDGE, M. 1998. *Agent Technology: Foundations, Applications and Markets*. Springer.

KHOSLA, R. and DILLON, T. 1997. *Engineering Intelligent Hybrid Multi-Agent Systems*. Kluwer Academic Publishers.

LU, Yong-Zai. 1996. *Industrial Intelligent Control*. John Wiley & Sons.

MATARIC, M. J. 1994. Learning to Behave Socially. In: *From Animals to Animats 3, Proceedings of the Third International Conference on the Simulation of Adaptive Behaviour*, Brighton. pp.453 - 462 MIT Press.

McCOY, A. W. and HOOPER, A. V. 1953. Blade Adjustment in Axial-Flow Compressor Stages. In: *Journal of the Aeronautical Sciences*, January 1953, pp.43-48.

McGUIRE, J., HUOKKA, D., WEBER, J., TENENBAUM, J., GRUBER, T., and OLSEN, G. 1993. SHADE: Technology for Knowledge-Based Collaborative Engineering. In: *Journal of Concurrent Engineering: Applications and Research,* 1(3).

MUSCATO, G. 1998. Soft Computing Techniques for the Control of Walking Robots. In: IEE *Computing & Control Engineering Journal*. August, 1998. Vol 9, No.4, pp.193-200.

NORMAN, T J., JENNINGS, N R., FARATIN, P. and MAMDANI E.H. 1993. Designing and Implementing a Multi-Agent Architecture for Business Process Management.
In: Muller J.P., Wooldridge, M.J., and Jennings, N.J. eds., *Intelligent Agents III: Agent Theories, Architectures, and Languages, Lecture Notes in Artificial Intelligence 1193*, Springer pp. 261-275.

O'HARE, G.M.P. and JENNINGS, N. R. 1996. *Foundations of Distributed Intelligence*. John Wiley.

OVERGAARD, L., PETERSEN, H. G. and PERRAM, J. W. 1994. Motion Planning for an Articulated Robot: A MultiAgent Approach. In: *Proceedings of Modelling Autonomous Agent in a MultiAgent World*, Odense University, pp.171-182

PADUANO, J. D., EPSTEIN, A. H., VALAVANI, L., LONGLEY, J. P., GREITZER, E. M. and GUENETTE, G. R. 1993. Active Control of Rotating Stall in a Low-Speed Axial Compressor. In: *Transactions of the ASME*, Vol. 115, January 1993, pp.48-56.

PARANUK, V.D., BAKER, A. and CLARK, S. 1998. The AARIA Agent Architecture: From Manufacturing Requirements to Agent-Based System Design. In: *Working Notes of the Agent-Based Manufacturing Workshop*, Minneapolis, MN.

RIESS, W. and BLÖCKER, U. 1987. Possibilities for On-line Surge Suppression by Fast Guide Vane Adjustment in Axial Compressors. In: *AGARD Conference Proceedings 421,* 69^{th} symposium of PEP, pp. 31-1 to 31-13.

RZEVSKI, G. 1994. *Designing Intelligent Machines Volume 1, Perception, Cognition and Execution*. Butterworth-Heinemann.

RZEVSKI, G. 1998. *Engineering Design for the Next Millenium: The challenge of artificial intelligence*. The 86th Thomas Hawksley Memorial Lecture IMechE, 9 December, London..

SARIDIS, G. N. 1979. Towards the realization of intelligent controls.
In: Proceedings of the IEEE, vol. 67, no.8. 1979.

SCHILDT, H. 1995. *Schildt's Windows 95 Programming in C and C++*. Osborne McGraw-Hill.

SHEN, W., XUE, D. and NORRIE, D. H. 1998. An Agent-Based Manufacturing Enterprise Infrastructure for Distributed Integrated Intelligent Manufacturing Systems.
In: *Proceeedings of PAAM'98*, London, UK, pp. 533-548.

SHEN, W., NORRIE, D. H. and BARTHES, J-P. 2001. *Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing,* Taylor and Francis.

SHEPHERD, D. G. 1956. *Principles of Turbomachinery,* Macmillan, New York.

SHOHAM, Y. 1993. Agent-oriented programming. In: *Artificial Intelligence.* **60** (1), pp 51-92. Elsevier Science Publishers B.V.

SUN, J. and ELDER, R. 1998. *Numerical Optimisation of Stator Vane Setting in Multistage Axial Flow Compressors*. Internal paper, School of Mechanical Engineering, Cranfield University, Bedford.

STEELS, L. 1994. A Case Study in the Behavior-Oriented Design of Autonomous Agents. In: *Proceedings of the 1994 Conference on the Simulation of Adaptive Behavior,* MIT Press.

STEINKE, R. J. 1982. *STGSTK A Computer Code for Predicting Multistage Axial-Flow Compressor Performance by a Meanline Stage-Stacking Method.*
NASA Technical Paper 2020.

WEISS, G. 1999. *Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence,* The MIT Press.

WITTIG, T. 1992. *ARCHON an architecture for multi-agent systems*.
Ellis Horwood Ltd.

WRIGHT, P. I. and MILLER, D. C. 1991. *An Improved Compressor Performance Prediction Model.* Conference Paper C423/028, IMechE., London..

APPENDIX

Compressor Flow Model

The compressor flow model used in the simulation program for the 'intelligent geometry compressor' represents the steady state performance of a hypothetical 5-stage machine with inlet guide vane (IGV). The model is intended to demonstrate changes in stage and overall performance parameters as a result of changes in stator angles and downstream throttle setting. The calculation procedure is based on the method proposed by Howell and Bonham (1950) for the prediction of single stage characteristics.

The stage prediction method combines the results of a simple 2-dimensional aerodynamic analysis with empirical factors and correlation data gained from a wide range of experimental work on cascades of blades and multi-stage machines. It enables the pressure rise across the stage and the efficiency to be calculated for a given range of flow values. The particular example given by Howell and Bonham is taken as the specification for a single stage in the compressor model. A hypothetical machine is then defined as a multiple of such stages each having the same operating point at design.

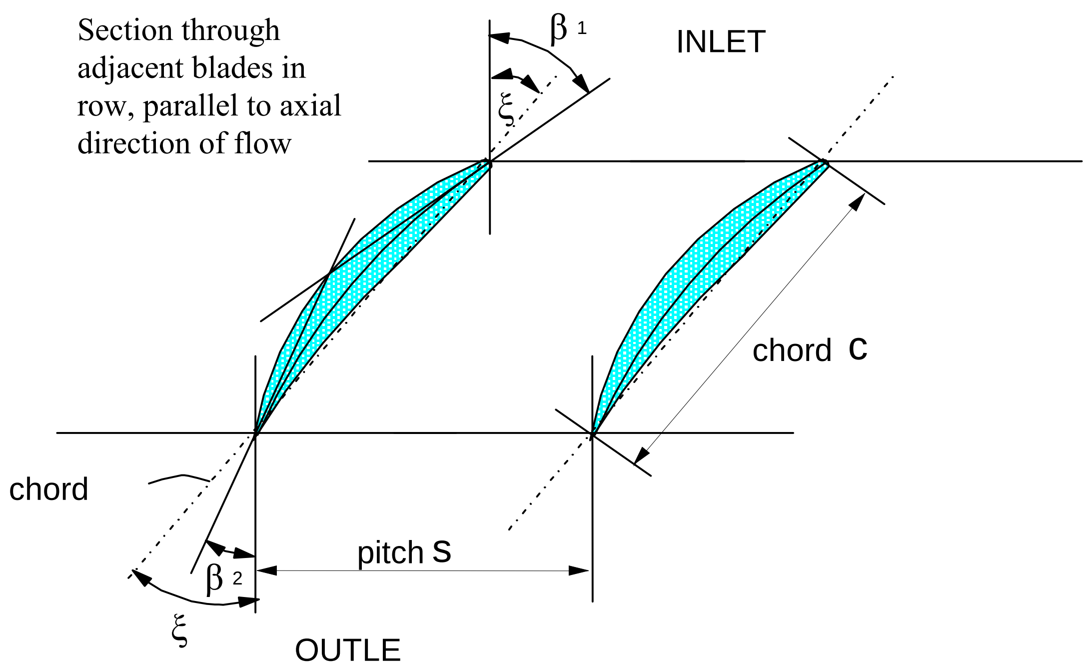
There are more recent methods in the literature for axial compressor performance prediction e.g. Steinke (1982), Wright and Miller (1991), but Howell and Bonham's work was selected because it is well established and widely referred to in standard texts on axial compressors e.g. Horlock (1958), Shepherd (1956) and Cumpsty (1989). In addition their numerical example, which was shown to agree closely with experimental results, provides a ready means of validating the calculation method as implemented in the simulation program.

The method of Howell and Bonham does not reflect the separate effects on stage pressure rise of rotor and stator explicitly but assumes a 50% reaction stage in which such effects are equal. To cater for variation in stator setting the pressure rise across rotor and stator must be treated separately and thus the 2-dimensional analysis is extended based on examples given in the texts by Horlock and by Shepherd.

A1. Definitions and Nomenclature

Stagger refers to the orientation of a blade or vane relative to the (rotational) axis of the machine. Usually, the stator setting is defined as the stagger angle but, given the reasonable assumption that the blade camber is based on a circular arc, then a change in stagger angle gives rise to a change in blade angle (outlet or inlet) of the same magnitude. Therefore, for present purposes, a change in stator setting will be reflected as a change in blade angle. The definitions of blade angles used in this analysis are given in Fig A1 below.

Fig A1 Blade Nomenclature



where:-

β_1 = blade inlet angle

β_2 = blade outlet angle

ξ = stagger angle

Note: In this analysis the term 'blade' is used to mean rotor blade or stator vane. Also the form and pitch of rotor blades is assumed to be the same for stator rows and IGV.

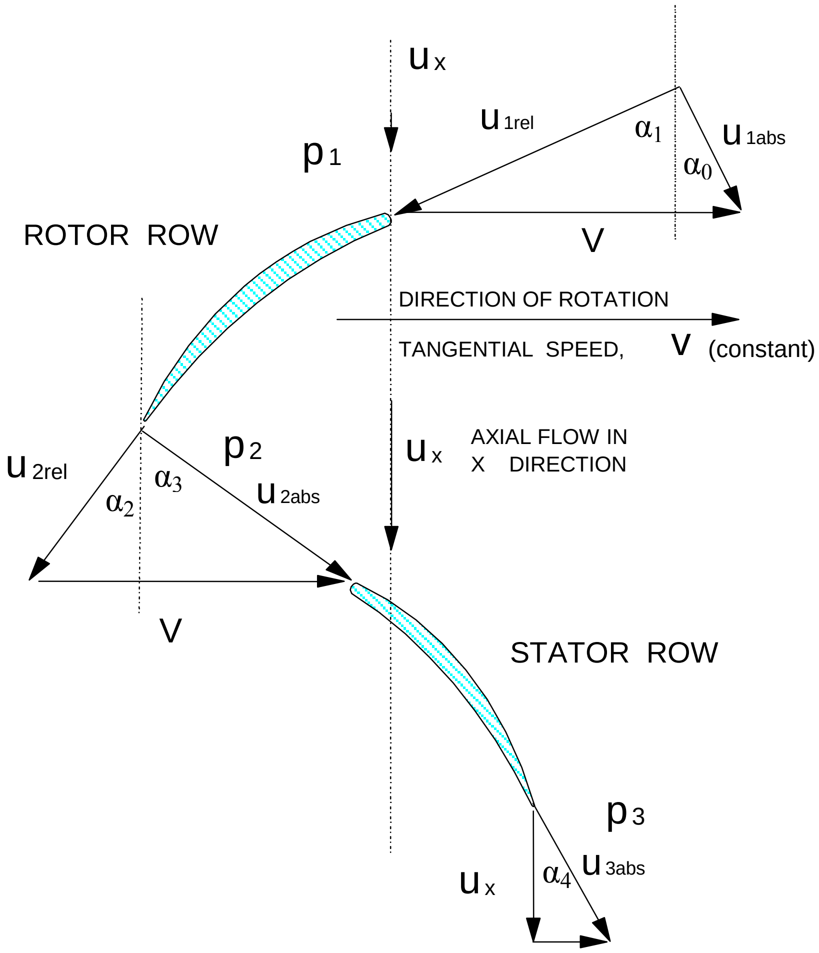
Other parameters used in the analysis are defined as follows:

u_{abs} = absolute flow velocity
u_{rel} = relative flow velocity
u_t = tangential flow velocity
u_x = axial flow velocity
V = rotor tip (tangential) speed
ϕ_x = axial flow function = u_x / V
p = absolute static pressure
p_d = absolute static delivery pressure
ρ = fluid density
Δ = pressure rise / loss
ψ = pressure rise coefficient = $\Delta /(1/2 \rho V^2)$
α_0 = absolute inlet flow angle to rotor
α_1 = relative inlet flow angle to rotor
α_2 = relative outlet flow angle from rotor
α_3 = absolute inlet flow angle to stator
α_4 = absolute outlet flow angle from stator
α_M = mean flow angle through blade row
$\in$ = fluid deflexion angle
i = incidence angle
(s/c) = ratio of pitch to chord
(h/c) = ratio of blade height to chord
C_D = total drag coefficient
C_L = coefficient of lift
η = efficiency
λ = 'work done' factor
R = stage reaction
D_k = rotor tip diameter (for stage k)
D_{in} = inlet section diameter
D_h = hub diameter
$\overline{D}_k$ = rotor tip diameter ratio = D_k/D_{in}
G = throttle setting coefficient

A2. Equations for Stage Characteristics

The characteristics of interest are pressure* rise (or pressure rise coefficient), efficiency and stage reaction versus flow function. These are obtained from equations largely derived from the velocity triangles for a rotor and stator pair at the mean blade diameter section as shown in Fig A2 with the assumptions that the flow is incompressible and axial velocity is constant through the stage. (* ***static*** pressure unless otherwise stated)

Fig A2 Flow through Stage



a) <u>Stage Pressure Rise</u>

Firstly consider flow to be reversible so that the ***ideal*** pressure rise across rotor due to momentum change only is given by:-

$$\Delta_{IR} = p_2 - p_1 = \frac{1}{2}\rho\left(u_{1rel}^2 - u_{2rel}^2\right)$$

The relative velocities can be expressed in terms of flow angles and the axial flow function, ϕ_x, so that:

rotor: $$\Delta_{IR} = \frac{1}{2}\rho V^2 \phi_x^2 \left(\tan^2 \alpha_1 - \tan^2 \alpha_2\right) \qquad \text{----------- 1)}$$

Similarly, the ideal pressure rise across the stator is given by:-

$$\Delta_{IS} = p_3 - p_2 = \frac{1}{2}\rho\left(u_{2abs}^2 - u_{3abs}^2\right)$$

and thus:-

stator: $$\Delta_{IS} = \frac{1}{2}\rho V^2 \phi_x^2 \left(\tan^2 \alpha_3 - \tan^2 \alpha_4\right) \qquad \text{----------- 2)}$$

To determine ***actual*** pressure rise across the rotor and stator, terms for pressure loss are introduced into the analysis. In addition, for consistency with Howell and Bonham, a 'work done' factor, λ, is also introduced which affords some correction for 3-dimensional flow effects at the blades. Thus actual pressure rise is given by:

rotor: $$\Delta_R = \lambda\left(\Delta_{IR} - \Delta_{LR}\right) \qquad \text{----------- 3)}$$

stator: $$\Delta_S = \lambda\left(\Delta_{IS} - \Delta_{LS}\right) \qquad \text{----------- 4)}$$

where:

Δ_{LR} = pressure loss across rotor
Δ_{LS} = pressure loss across stator

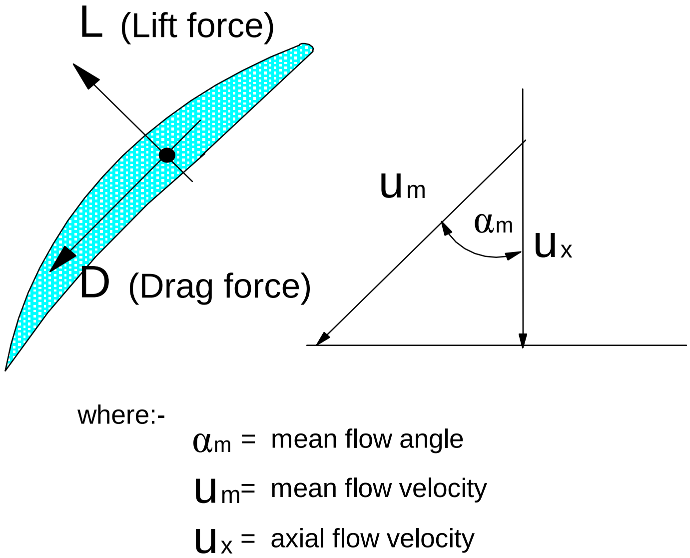
The actual pressure rise across the stage is then given by the sum of pressure rise across rotor and stator rows obtained from equations 3) and 4).

stage: $$\Delta_{Stage} = \Delta_R + \Delta_S$$

and stage pressure rise coefficient, $$\psi = \frac{\Delta_{Stage}}{\frac{1}{2}\rho V^2} \qquad \text{------------- 5)}$$

b) Pressure Loss across a Blade Row

The pressure loss across a blade row is expressed in terms of ***drag*** and the general form can be determined by analysis of the aerodynamic forces acting on a single blade with reference to the sketch below.



Equating pressure force to drag force yields:

for the rotor, $$\Delta_{LR} = \frac{1}{2}\rho \; V^2 \; C_{DR} \left(\frac{c}{s}\right)\frac{\phi_x^2}{\cos^3\alpha_{MR}} \qquad \text{---------} \;\; 6)$$

and for the stator, $$\Delta_{LS} = \frac{1}{2}\rho \, V^2 \; C_{DS} \left(\frac{c}{s}\right)\frac{\phi_x^2}{\cos^3\alpha_{MS}} \qquad \text{---------} \;\; 7)$$

where C_{DR} and C_{DS} are drag coefficients for rotor and stator respectively and the mean flow angles are found from:

$$\tan\alpha_{MR} = \frac{1}{2}\left(\tan\alpha_1 + \tan\alpha_2\right) \text{ and } \tan\alpha_{MS} = \frac{1}{2}\left(\tan\alpha_3 + \tan\alpha_4\right) \text{ ----------} \; 8)$$

Howell and Bonham incorporate corrections into the drag coefficient for other losses so that the definition of drag coefficient, generally, is given by:

$$C_D = C_p + C_a + C_s \quad \text{-------------} \quad 9)$$

where:

C_p = profile drag coefficient

C_a = annulus drag coefficient $\left(= 0.02\left(\frac{h}{c}\right)\right)$

C_s = secondary drag coefficient $(= 0.018\, C_L^2)$

C_L = coefficient of lift

A full explanation of the fluid effects which these coefficients represent may be found in Howell (1945).

For the coefficient of lift Howell and Bonham use an equation of the following form:

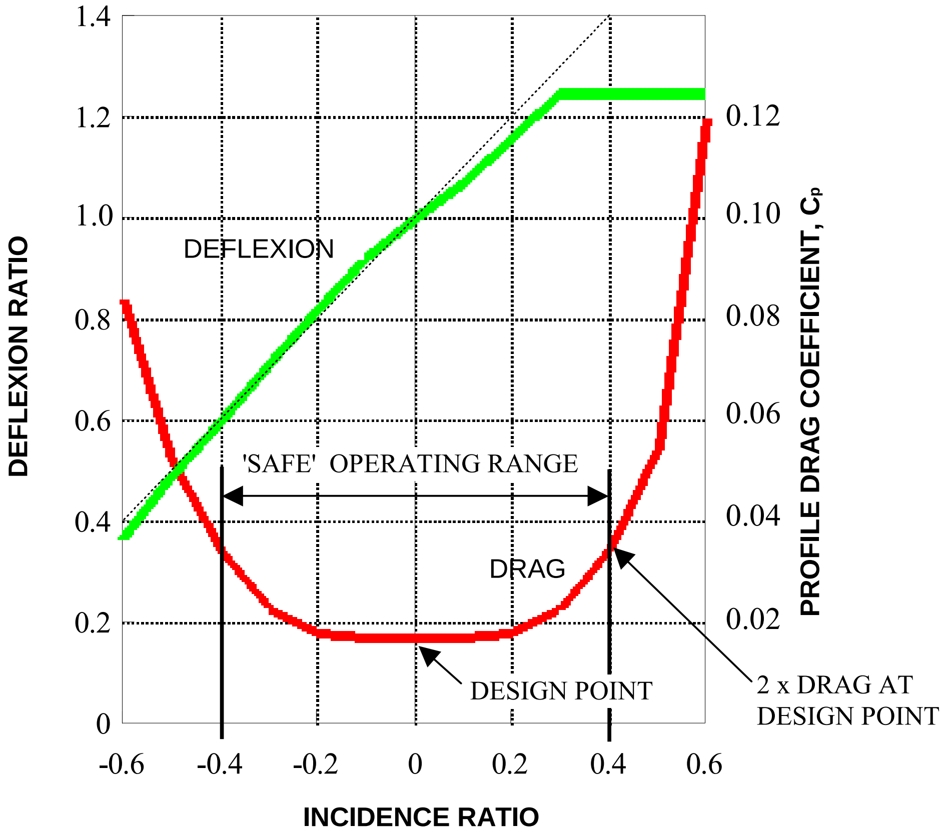
$$C_L = 2\left(\frac{s}{c}\right)(\tan\alpha_1 - \tan\alpha_2)\cos\alpha_M \quad \text{----------} \quad 10)$$

This can be applied to the rotor and stator by using the appropriate inlet, outlet and mean flow angles.

The profile drag coefficient depends on incidence and Howell and Bonham give empirical data in graphical form from which specific values of C_p can be found. This graph is reproduced in Fig A3 and gives curves of ***fluid deflexion ratio*** and ***drag coefficient*** against ***incidence ratio.***

Fluid deflexion $\in$ is the difference between inlet and outlet flow angles (e.g. $\alpha_1 - \alpha_2$) and incidence i is the difference between flow and blade angles at inlet (e.g. $\alpha_1 - \beta_1$). The deflexion ratio = $\in/\in^*$ and incidence ratio = $(i - i^*)/\in^*$ where * denotes values at design.

Fig A3 Howell and Bonham Empirical Data



The above data shows that drag effects increase significantly with change in incidence from the value at design. At some point, the drag effects will be sufficient to precipitate a local break down in the flow and some part or parts of the blade row will become stalled. As a guide for "safe" operation Howell and Bonham suggest that the profile drag coefficient should not be more than twice the value at design. This means that incidence ratio should be limited to the range ± 0.4. For present purposes this limit is taken to represent the point of stall onset and a ***static stall margin***, **sm**, variable is defined as:

$$sm = 1 - (\text{incidence ratio} / 0.4) \qquad \text{----------- } 11)$$

Thus at design, $sm = 1$ and at incidence ratio = 0.4, $sm = 0$.

c) Stage Efficiency

Stage efficiency, η_{stage}, is defined as:

$$\eta_{stage} = 1 - \frac{\Delta_L}{\Delta_I} \qquad \text{------------- 12)}$$

where $\Delta_L = \Delta_{LR} + \Delta_{LS}$ = sum of pressure losses through stage

and Δ_{LR} = pressure loss across rotor

Δ_{LS} = pressure loss across stator

d) Stage Reaction, R

Stage reaction is defined as the ratio of pressure rise across the rotor to the pressure rise across the stage i.e.

$$R = \frac{\Delta_R}{\Delta_{Stage}} \qquad \text{------------- 13)}$$

e) Howell and Bonham Equations for Stage Characteristics

The equations presented in the previous sections enable the rotor and stator to be treated separately. Therefore the resulting characteristics are able to reflect the effects of stator resetting i.e. different values of blade inlet angle. The equations given by Howell and Bonham are strictly valid for a 50% reaction stage for which the flow angles are related as follows: $\alpha_1 = \alpha_3$; $\alpha_2 = \alpha_4 = \alpha_0$

In this case the stage pressure rise coefficient is given by:-

$$\psi = \eta_{stage}\, 2\, \lambda\, \phi_x\, (\tan\alpha_1 - \tan\alpha_2)$$

and the stage efficiency by:-

$$\eta_{stage} = 1 - \left(\frac{2}{\sin 2\alpha_M}\right)\left(\frac{C_D}{C_L}\right)$$

It can be shown that these reduced equations as used in the Howell and Bonham prediction method are entirely consistent with the preceding analysis and can be derived from the more general equations presented earlier.

A3. Calculation of Stage Characteristics

The process starts with a value of flow function, ϕ_x, in the range of interest and then the pressure rise is calculated for the rotor and stator in turn.

a) Rotor

The absolute inlet angle of the flow into the rotor, α_0, is taken as being equal to the outlet flow angle of the upstream stator, α_4. The relative inlet flow angle, α_1, is then obtained from the rotor inlet velocity triangle as:-

$$\tan\alpha_1 = \frac{1}{\phi_x} - \tan\alpha_0 \qquad \text{--------------------- 14)}$$

Given knowledge of the flow angles at design then the incidence ratio can be found from:-

$$\left(\frac{i-i^*}{\in^*}\right)_{rotot} = \frac{(\alpha_1-\beta_1)-(\alpha_1^*-\beta_1^*)}{(\alpha_1^*-\alpha_2^*)} = \frac{(\alpha_1-\alpha_1^*)}{(\alpha_1^*-\alpha_2^*)} \qquad \text{--------------------- 15)}$$

since the rotor blade angle is fixed.

The deflexion ratio, $\left(\frac{\in}{\in^*}\right)_{rotor}$,corresponding to the incidence ratio is then obtained from Howell and Bonham's empirical data and hence the rotor outlet flow angle, α_2, from:-

$$\alpha_2 = \alpha_1 - \in^*_{rotor}\left(\frac{\in}{\in^*}\right)_{rotor} \qquad \text{--------------------- 16)}$$

(Note: For purposes of calculation the empirical data is applied by means of a set of discrete values with linear interpolation.)

The profile drag coefficient is also obtained from the empirical data so that, with the flow angles known, the pressure loss across the rotor can be calculated from equations 6), 8), 9) and 10). Finally, the actual pressure rise across the rotor is found using equations 1) and 3).

b) Stator

A similar procedure is applied to the stator. The steps are summarised below.

inlet flow angle:-

$$\tan\alpha_3 = \frac{1}{\phi_x} - \tan\alpha_2 \qquad \text{---------------------17)}$$

incidence ratio:-

$$\left(\frac{i-i^*}{\epsilon^*}\right)_{stator} = \frac{(\alpha_3-\beta_1)-(\alpha_3^*-\beta_1^*)}{(\alpha_3^*-\alpha_4^*)} = \frac{(\alpha_3-\alpha_3^*)-\delta\beta_1}{(\alpha_3^*-\alpha_4^*)} \qquad \text{------------- 18)}$$

where $\delta\beta_1$= change in stator blade inlet angle (≡ stagger adjustment).

deflexion ratio:-

$$\left(\frac{\epsilon}{\epsilon^*}\right)_{stator} \Leftarrow \text{Howell and Bonham's empirical data} \Leftarrow \left(\frac{i-i^*}{\epsilon^*}\right)_{stator}$$

stator outlet flow angle:-

$$\alpha_4 = \alpha_3 - \epsilon^*_{stator}\left(\frac{\epsilon}{\epsilon^*}\right)_{stator} \qquad \text{--------------------- 19)}$$

profile drag coefficient:- $\Leftarrow$ Howell and Bonham's empirical data $\Leftarrow \left(\frac{i-i^*}{\epsilon^*}\right)_{stator}$

pressure loss across the stator:- $\Leftarrow$ equations 7), 8), 9) and 10).

pressure rise across the stator:- $\Leftarrow$ equations 2) and 4).

c) Stage

With rotor and stator variables known, values for stage pressure rise, efficiency and reaction are found from equations 5), 12) and 13). The whole process is repeated for each value of flow function, ϕ_x, in the range of interest to generate complete characteristics.

A4. Calculation for Multiple Stages

The basic approach for multiple stages is to apply the method of the previous section to calculate stage performance and then to re-calculate the fluid density and axial flow function before the next (downstream) stage calculation in order to take account of pressure and flow area changes. For this purpose, adiabatic compression is assumed and applied in conjunction with continuity of mass flow.

The relevant equations for the k[th] stage are:-

$$\phi_{xk} = \frac{\left(\dfrac{\rho_0}{\rho_k}\right)\phi_{x_0}}{\overline{A}_k \ \overline{D}_k} \qquad \text{----------------} \quad 20)$$

$$\rho_k = \rho_0 \left(\frac{p_k}{p_0}\right)^{\frac{1}{\gamma}} \qquad \text{----------------} \quad 21)$$

Note that ρ_0, p_0, and ϕ_{x0} refer to inlet flow conditions of the compressor and $\overline{A}_k$ *and* $\overline{D}_k$ are sectional flow area and rotor tip diameter ratios referred to the inlet section of the machine. Also p_k is the absolute static pressure at inlet to stage k.

For n stages, the overall pressure rise is the sum of the actual pressure rise for each stage of the machine,

$$\Delta_{overall} = p_d - p_0 = \sum_{k=0}^{k=n} (\Delta_{Stage})_k \qquad \text{----------------} \quad 22)$$

where p_d and p_0 are the absolute static pressures at the outlet of the last stage (i.e. delivery pressure) and at the inlet of the inlet guide vane respectively. (Note: stage k=0 represents inlet guide vane which is treated as a "rotor-less" stage).

The overall efficiency is then the ratio of actual to ideal overall pressure rise,

$$\eta_{overall} = \frac{\Delta_{overall}}{\sum_{k=0}^{k=n} \left(\Delta_{IR} + \Delta_{IS}\right)_k} \qquad \text{----------------} \quad 23)$$

The process is repeated for a range of values of axial flow function to generate the complete overall characteristics.

A5. Hypothetical Machine Specification

The hypothetical machine comprises five stages and an inlet guide vane (IGV) and the working fluid is air. The stages are geometrically similar and the form of the blades identical for all rows in the machine. In addition, the stagger angles of all stator rows and IGV are variable. A nominal or design setting is specified based on the geometry of the 50% reaction stage given in the example presented by Howell and Bonham. Thus at design it is required that the flow function for each stage, $\phi_{x,}$ is 0.837 and the corresponding stage pressure rise coefficient, ψ, is 1.025. The overall static pressure rise at design is chosen to be 4.0 bar. An additional requirement is that the total number of stator and IGV blades is about 100 with a maximum of 20 in a row.

a) Flow and stagger angles at design

The flow angles for all stages at design are adopted directly from Howell and Bonham:

$$\alpha^*_1 = \alpha^*_3 = 43.9°; \text{ and } \alpha^*_2 = \alpha^*_4 = \alpha^*_0 = 13.1°$$

The corresponding nominal stator blade outlet angle, β, (taken as the stagger angle) is the same for all stages and is calculated to be 5.6° from the following equation:-

$$\beta = \frac{(\alpha_4^* - 7.1554)}{1.0622} \qquad \text{-------------------- 24)}$$

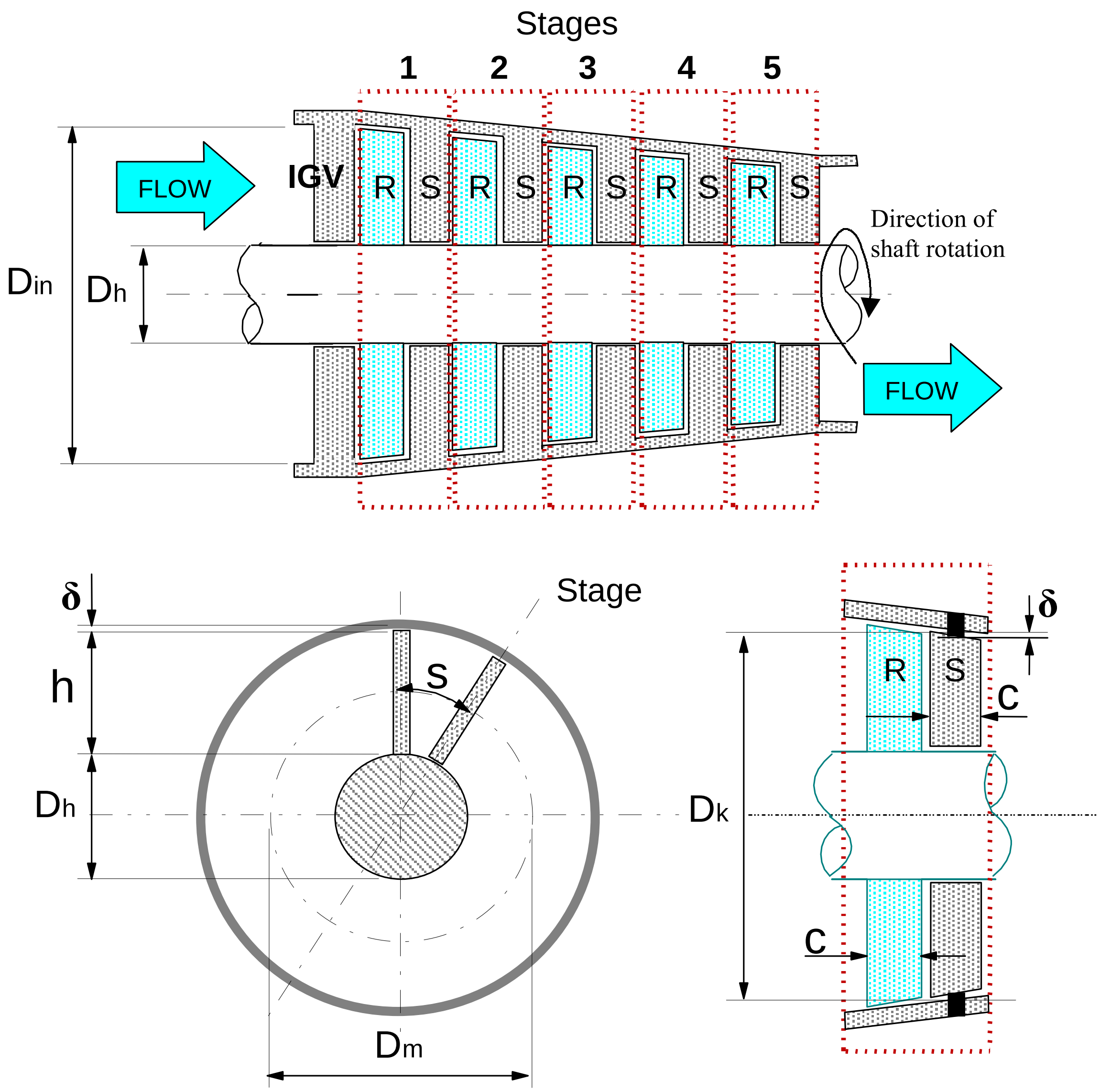
This equation is based on an empirical relationship given by Howell and Bonham to correct the outlet flow angle of the stator for flow deviation.

The inlet flow angle to the IGV is assumed to be zero, (i.e. inlet flow is parallel to the axis of the machine), and constant, independent of axial flow velocity. It will be noted that this assumption means that the incidence ratio for the IGV (and hence stall margin) is determined only by the stagger setting for the IGV. The stagger angle and outlet flow angle of the IGV at design are the same as for stators i.e. 5.6° and 13.1° respectively.

b) Dimensions and rotational speed

The configuration of the hypothetical machine and the principal dimensions of interest

Fig A4 Hypothetical Machine



$\mathbf{D_{in}}$ = inlet section diameter

$\mathbf{D_h}$ = hub diameter

$\mathbf{D_k}$ = rotor tip diameter of stage 'k'

$\mathbf{D_m}$ = diameter at mean blade height of stage 'k'

h = blade height

s = pitch at mean diameter

c = blade chord

δ = radial clearance at root of stator vane to allow stagger adjustment are defined in Fig A4. Based on initial approximate calculations a rotational speed of 6000 rpm and hub diameter of 0.3m were selected. An initial value of 1.0m was adopted for the inlet section diameter, D_{in}. This is also the value for the IGV outer diameter and the rotor tip diameter of the first stage. An iterative process of stage calculation was then followed to arrive at values of rotor tip diameter ratio, $\overline{D}_k$, which satisfy the stage and overall performance requirements at design. Dimensions were finally adjusted to meet the constraints on the number of stator and IGV blades using the following approximate equation:-

$$\text{number of blades in row,} \quad m = \frac{\pi\, D_m}{s} = 8.4679\left(\frac{\overline{D}_k + \overline{D}_h}{\overline{D}_k - \overline{D}_h}\right) \quad \text{-----------} \quad 25)$$

This equation is derived from the stage geometry shown in Fig A4 and the relationships connecting pitch, blade height and chord given by Howell and Bonham

i.e. s = 0.742 c; and h = 2 c.

The dimensions finally adopted for the hypothetical machine are given below.

	Rotor Tip Diameter Ratio ($\overline{D}_k$)	Number of blades in stator row
IGV	1.0000	16
Stage 1	1.0000	16
Stage 2	0.8947	17
Stage 3	0.8250	18
Stage 4	0.7740	19
Stage 5	0.7347	20

Inlet section diameter = 1.03 m
Hub diameter = 0.3 m
Rotational speed = 6000 rpm

c) Stagger angle range

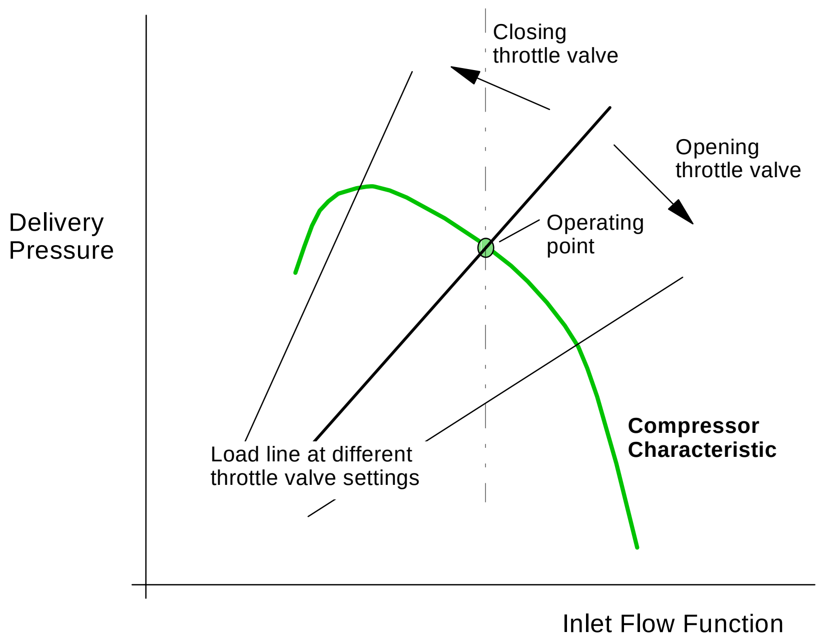
The amount of adjustment of stagger angle for stator and IGV is ultimately limited by mechanical contact between the blades and the inside of the compressor casing. There must be sufficient radial clearance, δ, between the root of the stator blades and the casing to allow for movement of the blades. An approximate expression for the maximum stagger angle can be derived from the geometry involved as:-

$$\beta_{\max} = \pm \sin^{-1}\left[\frac{8\sqrt{\left(\frac{\delta}{D_k}\right)-\left(\frac{\delta}{D_k}\right)^2}}{\left(1-\left(\frac{D_h}{D_k}\right)-2\left(\frac{\delta}{D_k}\right)\right)}\right] \qquad \text{-----------} \quad 26)$$

Based on the stator of stage 5 where the case inner diameter is smallest a maximum range of stagger angle of about ± 35° is possible for a radial clearance of about 1.5 mm.

A6. Calculation of Operating Point

The overall operating point for the hypothetical compressor is defined by the intersection of the pressure-flow characteristic with the load line of the downstream throttle valve as shown in the sketch below.



The throttle characteristic represented by the load line is assumed to be linear such that:-

Delivery pressure, $$p_d = \frac{\phi_{x0}}{G}$$

where:
ϕ_{x0} = flow function at the inlet to the compressor and is proportional to mass flow rate through the machine
G = 'conductance' coefficient and represents the internal flow area of the throttle

The compressor characteristic is firstly calculated and stored as a set of p_d and ϕ_{x0} values. The point of intersection with the throttle load line is then found by a process of linear interpolation. This defines the overall operating point for the particular stator and throttle settings that apply. Once the inlet flow function at the overall operating point is known then the operating points of individual stages may be found, again, by linear interpolation.

A7. Summary of Calculation Procedure for Hypothetical Machine

The calculation methods described so far form the basis of the compressor flow model used in the simulation programs. The flow chart of Fig A5 summarises the overall calculation procedure as embodied in the computer code.

A8. Validation of Calculation Implementation

To confirm correctness of the compresor flow model implementation various runs of the simulation program were made at design and the results compared with those given by Howell and Bonham. Exact agreement within the bounds of calculation accuracy was noted at every point of comparison as illustrated in the sample stage characteristics shown in Fig A6. Finally, the results obtained for all stages at design and overall performance are summarised in the table of Fig A7.

.

Fig A5 Summary of Calculation Procedure

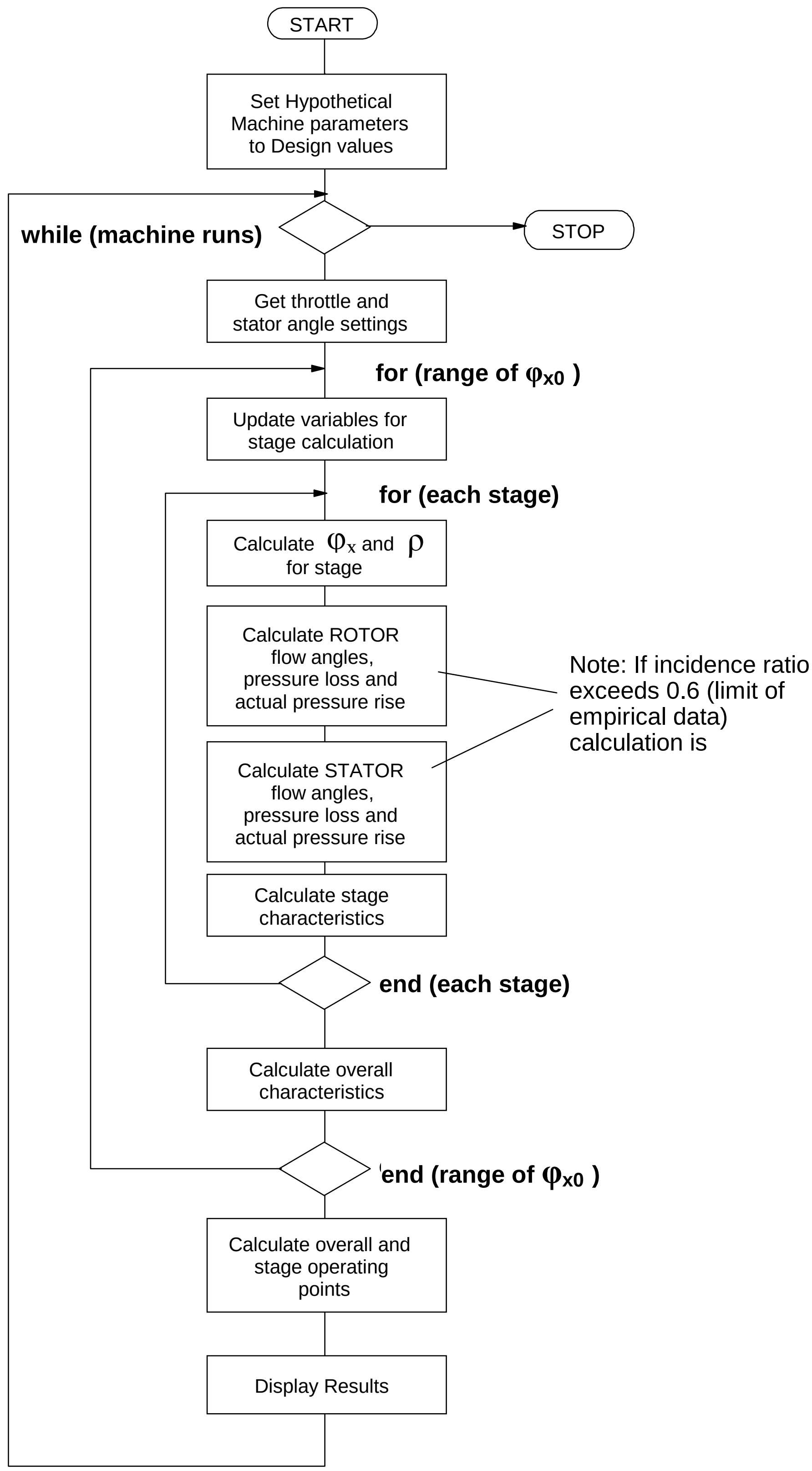
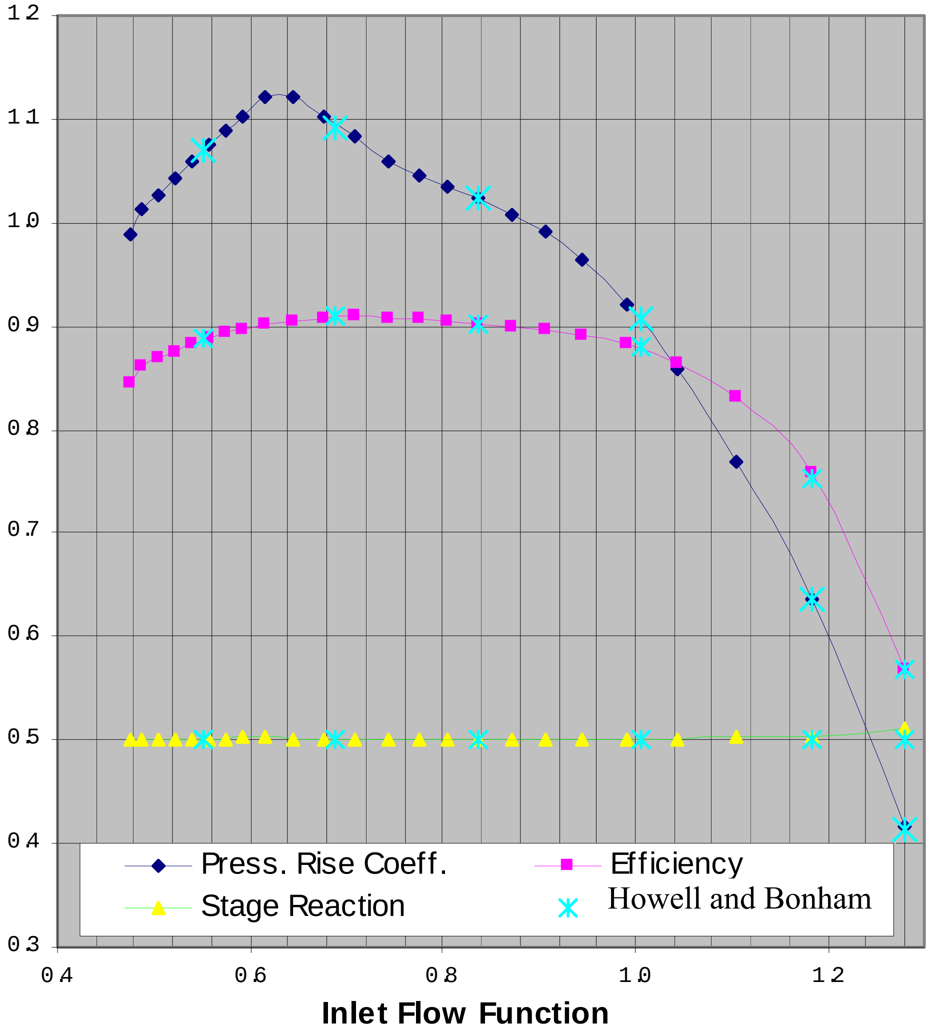


Fig A6 Sample Calculated Stage Characteristics



Note:

Graphs show calculated data points compared to those given by Howell and Bonham (1950). The calculated results are those for stage 5 of the hypothetical compressor obtained using the simulation program IGC1. This stage is the furthest downstream stage and most closely approximates an isolated stage for which the Howell and Bonham results are strictly valid.

Fig A7 Hypothetical Machine Performance at Design

Calculated Results

	STAGE GEOMETRY						PERFORMANCE					
	FLOW ANGLES					ROTOR DIAMETER RATIO	FLUID DENSITY RATIO	EFFICI-ENCY %	FLOW FUNCTION	PRESS. RISE COEFF.	STATIC PRESSURE AT OUTLET bara	STAGE REACTION %
	α_0	α_1	α_2	α_3	α_4	$\overline{D}$	$\overline{\rho}$	η	φ_x	ψ	p	R
VARIABLE INLET GUIDE VANE (IGV)	-	-	-	0.000	13.104	1.0000	1.000	-	0.816	-	0.977	-
Rotor **STAGE 1**	13.104	43.877	13.112			1.0000	0.975	90.2	0.837	1.024	1.617	50
Variable Stator				43.873	13.109							
Rotor **STAGE 2**	13.109	43.880	13.112			0.8947	1.393	90.2	0.837	1.024	2.349	50
Variable Stator				43.878	13.112							
Rotor **STAGE 3**	13.112	43.897	13.117			0.8250	1.816	90.2	0.837	1.024	3.161	50
Variable Stator				43.894	13.116							
Rotor **STAGE 4**	13.116	43.877	13.112			0.7740	2.241	90.2	0.837	1.024	4.042	50
Variable Stator				43.877	13.113							
Rotor **STAGE 5**	13.113	43.871	13.112			0.7347	2.668	90.2	0.837	1.024	4.990	50
Variable Stator				43.871	13.104							
OVERALL PERFORMANCE AT DESIGN								89.9	0.816	6.019	4.990	-

Notes

1) Blade outlet angle (stagger) for IGV and stator rows = 5.6 degrees

2) Flow angles and performance of each stage agree with example of Howell and Bonham within calculation accuracy.

3) Overall efficiency less than stage efficiency due to pressure loss at IGV