CRANFIELD INSTITUTE OF TECHNOLOGY

COLLEGE OF AERONAUTICS

AIRCRAFT DESIGN DIVISION

Ph.D. Thesis

Academic Year 1986-87

J. ALSINA

Development of an Aircraft Design Expert System

Supervisor: Prof. A.J. MORRIS

April 1988

SUMMARY

The aircraft design process is characterised by the application of a wide range of knowledge across many disciplines based upon a certain degree of judgement and experience of the designer. A two pass approach has been taken towards the development of an aircraft design expert system based on the requirements of two conceptually different design steps namely, wing design and aircraft configuration. The current status of the work is one where an actual program for wing design exists with supporting documentation, and a very effective examination of the knowledge base performed based on the detail investigation of overall aircraft design process with particular emphasis on the wing design and the aircraft configuration design steps.

The approach taken accomplishes the objectives of the current research in defining the knowledge base, providing tools and specifications for tools to be used within an aircraft design expert system closely following the problem-solving techniques utilised by the design expert.

ACKNOWLEDGEMENTS

The author wishes to express his sincere appreciation to Prof. Alan Morris for obtaining the ESPRIT contract under which this work has been undertaken and for the many useful discussions during the course of this thesis and for his constant support, encouragement and friendship.

My thanks are due to Dr. John Fielding for the many useful discussions during the course of this thesis, and for his constant encouragement and friendship.

My very special thanks to Emma Barton for her constant support and encouragement during the course of this work.

CONTENTS

CHAPTER 6 TRIAL IMPLEMENTATION

FIGURES

Figures		Page

NOTATIONS

η	2y/b
ϵ_o	angle of twist at wing tip
ϵ	angle of twist (distribution)
α_o	overall zero lift angle for the wing
λ	taper ratio
Λ1/4	quarter chord wing sweep back angle
a_o	average lift curve slope for the whole wing
A	aspect ratio
b	wing span
$\bar{c}$	mean wing chord
c(y)	wing chord at position y
C	constant depending on aircraft type
$\overline{CL}$	cruise lift coefficient
CL(y)	local lift coefficient at position y
CLmax	maximum landing lift coefficient
CLs	stall lift coefficient
d	parameter difference
Decr	decrement due to fuselage-wing interaction
H	cruise altitude
i	section inportance (rating)
K	correction factor
m	$1 - M^2$
M	Mach number

Mecon	economic cruise Mach number
Mmax	maximum cruise Mach number
MD	3-D drag rise Mach number
MD1	3-D drag rise Mach number for best section
MD2	3-D drag rise Mach number for second best section
$(MD)_{\Lambda=0}$	2-D drag rise Mach number
N	ultimate load factor
rd	parameter relative difference
ri	parameter relative importance
R	operating range
S	wing area
t/c	thickness/chord ratio
(t/c)1	thickness/chord ratio for best section
(t/c)2	thickness/chord ratio for second best section
(t/c)a	average thickness/chord ratio
(t/c)r	root thickness/chord ratio
T	atmospheric temperature
Vsound	speed of sound
VD	aircraft design diving speed
Ww	wing mass
W	aircraft maximum take-off mass
W/S	wing loading
y	semi-span position
$\bar{y}$	spanwise position of the centre of pressure

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The design process involves a series of artifact descriptions at various levels of detail. Typically, a search towards a solution does not follow a straight step by step path but may require a number of iterations through various parts of the process and the problem is seldom so simple or the designer so lucky that the first solution is optimal from the viewpoint of production, cost, servicing etc.

Solving a design problem requires both general knowledge about the domain and specific knowledge about the problem. It is an important human activity which is still poorly understood. From the psychological point of view it is interesting as a kind of complex problem-solving activity. From the engineering point of view it is interesting as a way of improving the cost and reliability of the design.

Aircraft design (1, 2, 3, 4, 5, 6) falls within this complex and cyclic process where new designs take a long time to evolve typically, 15 years for an airliner (7). Such long gestation periods without the advantage of overlapping projects causes serious problems within the aircraft industry. First of all, because of personnel changes each new design may be generated by, essentially, a novice design team which must somehow acquire the knowledge left within the organisation by earlier design teams. Secondly, a modern aircraft incorporates a high degree of novelty in materials, construction, analysis etc., from its predecessors which have to be assessed and included in the design by non-specialist design engineers. As a result there is a clear requirement that programs exist which preserve the knowledge base of the individual firm and also contain the information which will facilitate the incorporation of new technology. Reducing the development period will allow, for example, the use of electronic technology which is state-of-the-art a few years before the aircraft is completed, rather than technology which is close to being obsolete by the time the aircraft is in production.

Within computer science the field of Artificial Intelligence (AI) has been concerned for more than two decades (8, 9, 10, 11, 12) with building computational models of processes that would be considered intelligent if done by a human. Expert systems (13, 14, 15, 16), an application area of AI, has investigated methods and techniques for constructing computer programs with specialised problem-solving expertise. These programs have generated considerable interest within academia and industry as a result of the high levels of performance obtained in narrow problem areas where only experts used to perform and by providing a number of important benefits:

1. They provide a model in which the reasoning process employed by the expert in solving domain problems is encapsulated and can be extended and modified to include new developments and requirements.

2. The resulting system can be used to solve problems, freeing the expert to do more creative work and is a willing expert, able to eliminate errors due to oversight or fatigue.

3. They serve as teaching tools, in which the user is able to trace the expert system's reasoning and learn from it.

4. The exploitation of expert systems provides a formalisation and clarification of knowledge that results from the human expert making his reasoning explicit, and the possibility of combining the expertise from many human experts to create an optimum system, eliminating individual bias.

5. Finally, they offer the possibility of creating reasoning tools which can be used in similar domains.

Taking into account the current needs in aircraft design and the benefits which the development of an expert system for aircraft design could bring to this complex field, the specific objective of the present work has been the development of better models for the aircraft design process. Towards this goal, the aircraft design knowledge has been defined, as it is of paramount importance in structuring the inference and representation formalisms of the eventual expert system. Following a study of the aircraft design process, it was concluded that a full implementation of an expert system for aircraft design would not be appropriate, based on the current knowledge about the process. The approach taken - which is well suited to the present research objectives - was to develop some initial ideas with an implementation on a restricted part of the total design problem and then increase the level of complexity.

The wing design was selected as the task to be analysed and implemented during the first pass as it is a small and representative part of aircraft design which can be separated from other design steps without making too rigid assumptions about the final design. A detail study of the wing design knowledge base was performed (17) and a framework for an aircraft design expert system was defined (18). Within this framework the template concept was used to represent the wing design knowledge as a series of small and independent design steps. These templates were organised as a partially ordered tree to represent subgoal-supergoal, sequencial, and conjuctive relationships.

Based on this initial study, an attempt was made at modelling the wing design knowledge using an existing expert system shell namely, SAGE (19, 20). SAGE was used to model the problem of selecting a two-dimensional wing section from a choice of three as part of the wing design process. The resulting output in the form of a computer program (21) was crude and inflexible, while the task of 'formatting' the wing design knowledge into the SAGE language was rigid, repetitive, and odd in many cases. From this initial study, various control and man-machine interface requirements were laid down (21) and used during later implementations of the wing design task in the logic based PROLOG language (22, 23, 24) which provided the necessary flexibility and control structure to model the wing design knowledge.

A domain independent controller (25) was developed to provide the necessary design flexibility and correct execution sequence based on the evaluation status of the design. The features provided by this controller included:

1. Computes a list of executable steps based on the steps which have not been done and whose predecessors have all been done. This list can be ordered according to their executable order of preference.

2. Allows special values (commands) can be entered.

3. If the results of a step are not satisfactory the program allows the user to decide which steps have to be re-evaluated.

4. Allows the user to traverse the design tree in various directions depending on the status of the design.

This controller was evaluated from which a second controller implemented (26) with the following extensions:

1. Allows intelligent backtracking by computing the steps which might be the sources of the problems with the design. If there are not defined problem sources for a design step the program asks the user for the possible problem names and problem sources.

2. Several ways of computing any step are allowed.

The outcome of this first pass delivered a program (27, 28, 29) for wing design called ADROIT (Aircraft Design by Regulation of

Independent Tasks) together with an aircraft design expert system architecture. ADROIT designs a high subsonic aircraft wing by selecting a two-dimensional aerofoil section from a choice of three alternatives and evaluating a range of suitable sweep angles according to the aircraft specification. The above controller together with a powerful man-machine interface makes ADROIT easy and flexible to use allowing many wing design configurations to be tried quickly and effectively.

For the second pass towards the development of an aircraft design expert system, the configuration problem was chosen for analysis and implementation in order to increase the level of complexity. The success of a design largely depends on the aircraft configuration i.e., the general layout, external shape, dimensions etc., as characterised by the relative location of the main aircraft components i.e., wings, engines, tail surfaces, and undercarriage. It is based on the investigation into and interpretation of the aircraft function and a translation of the most pertinent requirements. A detailed study of the knowledge base (30, 31, 32) was performed which identified the sources of knowledge used by the expert designer during a solution, and classified this knowledge according to a specific type. An extensive enumeration of examples within each knowledge type was performed for reference during a later implementation. From this detailed study of the knowledge base, it became apparent that the framework proposed during the first pass would need to be modified or a completely new approach taken in order to effectively represent and control the aircraft configuration process. Two main conceptual differences between aircraft configuration and wing design were identified:

1. The size and complexity of aircraft configuration makes it difficult to identify in advance every possible design sequence or strategy to follow.

2. The smaller design steps into which the aircraft configuration process can be broken down are not independent, and the iterations which occur between these sub-steps are complex and difficult to identify in advance.

Based on these conceptual differences and the requirements possed by the nature of the aircraft configuration knowledge the modifications and extensions to the proposed framework were identified.

The remaining two sections of this Chapter introduce AI and outline the presentation of the work carried out during this research. The introduction to AI is performed by describing typical applications and knowledge representation techniques used in order to show the general computational concepts involved.

1.2 ARTIFICIAL INTELLIGENCE

Many human mental activities such as writing computer programs, doing mathematics, diagnosing a patient, understanding language, and even driving a car are said to demand intelligence. Most of the computer programs writing to carry out these tasks have taken place in the field called AI. As with any new field there are different views and beliefs of what AI is, a popular but perhaps limited definition of the field is interpreted from the Turing's test (10):

An interrogator is separated from a person (or machine) under interrogation, and communication is only possible using a terminal. The idea is that if the human cannot tell, through the interrogation, whether the communication is with another person or a machine, then the machine - if indeed it is a machine giving the answers - may be regarded as intelligent.

1.2.1 Applications

The AI application areas are very extensive thus, the enumeration is limited to showing the general computational concepts involved, the kinds of data structures used, the types of operations performed on these data structures, and the properties of control strategies used.

1.2.1.1 Problem-solving - Several rather distinct ways have emerged for the representing and thinking about problem-solving (33, 34, 35). One is to view problem-solving as a search in which a solution space is postulated together with legal 'moves' that alter this space. Thus, solving a problem consists in searching the model of the space (selectively) until a goal is encountered. A second way of viewing problem-solving is as reasoning in which the knowledge is represented in order to allow the deduction of new statements from axioms and previously deduced statements. Here, solving the problem consists in accumulating more and more information by inference until the answer to the problem has been found. In the third way a set of objects and various subsets are defined by the constraints they satisfy. Here, solving a problem consists in narrowing down the original set to a subset or unique object that satisfies all constraints. These types are not mutually exclusive but rather can be viewed as a combination of strategies.

Various computer programs have yielded research conclusions towards the development of intelligence in artificial systems. Game playing, for example, can enlarge human knowledge of the broad heuristic techniques that are necessary in solving economic, social and other problems in conditions that are constantly changing and difficult to define. Today's programs play championship level drafts and backgammon, as well as very good chess. The latter still presents problems since the computer is unable to see the board in terms of meaningful patterns and humans often solve a problem by finding a way of thinking that makes the solution easy whereas AI programs (so far) must be told how to think about the problems they solve.

1.2.1.2 Intelligent retrieval from databases - Database systems are computer programs that store a large body of facts about some subject in such a way that they can be used to answer questions about that subject.

Many techniques have been developed to enable the efficient representation, storage, and retrieval of large number of facts. From the AI point of view, the subject becomes interesting when there is a need to retrieve answers that require deductive reasoning with facts in the database. Problem areas in this application are understanding queries stated in natural language, how to deduce answers from the stored facts, and the representation of commonsense knowledge in order to understand a query and deduce an answer.

1.2.1.3 Natural language - When humans communicate with each other using language, they employ, almost effortlessly, extremely complex and still little understood processes. The evolution of language has exploited the fact that humans use their considerable computational resources and shared knowledge to generate and understand highly condensed and streamlined messages. Thus, generating and understanding language is a problem of fantastic complexity.

Some progress has been made towards understanding spoken and written fragments of languages. Typical applications include answering questions posed in English, translating sentences from one language to another, following instructions given in English, acquire knowledge by reading textual material, etc. Of great importance in all of these applications is the role of expectations and the representation of commonsense knowledge about the world.

1.2.1.4 Automatic programming - AI has investigated systems that can write computer programs from a description of what the program is to accomplish, automatic debugging of programs and programs that learn by modifying their own code. Some of the techniques used have been through examples, high-level language descriptions, and English algorithms.

1.2.1.5 Learning - A definition of learning (36) is that learning is any change in a system that allows it to perform better the second time on repetition of the same task or on another task drawn from the same domain. Very little progress has been made in learning since human intelligence is still poorly understood but, attempts have been made with programs (8) that learn from examples, their own experience, and from being told.

1.2.1.6 Expert systems - on first observation do not appear greatly different from equivalent computer programs which evaluate and define problems. However, the special feature which make these programs different is the representation and use of knowledge bringing important benefits to the modelling of the application domain as described above. Typically the user interacts with an expert system as he would interact with an expert i.e., explaining his problem, performing tests, and asking questions about proposed solutions. Current systems (37, 38) have achieved high levels of performance in consultation tasks but, are of limited scope and, unlike humans, do not know when they might be wrong.

The structure of expert systems

The ideal expert system has been described (13) as having the following components:

1. A language processor to mediate information exchanges between the expert system and the human expert.

2. A blackboard to record the intermediate results.

3. A knowledge base where the facts, heuristic planning and problem-solving rules are stored.

4. A scheduler to control the order in which the rules are processed.

5. An interpreter to apply the rules.

6. A consistency enforcer to adjust previous conclusions when new data or knowledge is collected.

7. A justifier to explain the system's behaviour.

Of these, the knowledge base, database, and interpreter are considered to be the essential parts of an expert system but as yet, there is no expert system with all the above components which can be applied to the solution of significant real world problems.

Expert system tools

Expert system tools are programming systems that simplify the job of constructing an expert system (39, 40). They range from very high-level programming languages to low-level support facilities and have been classified (39) as programming languages, knowledge engineering languages, system building aids, and support facilities.

The programming languages used for expert system are of two types:

1. Problem oriented languages such as FORTRAN and Pascal designed for particular classes of problems for example, FORTRAN has convenient features for performing 'number crunching'.

2. Symbol manipulation languages such as LISP and PROLOG designed for AI applications for example, PROLOG has mechanisms for manipulating symbols in the form of list structures to represent various objects.

Programming languages, like PROLOG, offer the greatest flexibility to the expert system builder or knowledge engineer but fail to provide guidance on how to represent knowledge or mechanism for accessing the knowledge base. On the other hand, knowledge engineering languages offer representation guidelines and controlling strategies but with little possibility of modifying or extending the control scheme.

Few system building aids currently exist, and range from those that help acquire and represent the domain expert's knowledge to those that help design the expert system under consideration.

Various support facilities have been developed for helping with programming such as debugging aids, knowledge base editors, incremental compilers, etc., and tools that enhance the capabilities of the finished system such as built-in input/output and explanation mechanisms.

Applications of expert systems

The enumeration is based on showing the general computational concepts involved, the kinds of data structures used, the types of operations performed on these data structures, the properties of control strategies used, and the relative success of each expert system.

R1 or XCON (41) is a rule-based expert system that configures VAX computers by determining the physical layout and interconnection of their many components from a customer's order. The diagrams produced are used by a technician to physically assemble the system. It is used routinely by Digital Equipment Corporation and over a period of 3 months in which 3000 orders were processed, 85 percent of these were found correct. When errors occurred it was because R1 lacked the information on recently introduce products, or due to known, correctable problems with the rules.

PROSPECTOR (42) assesses the suitability of a site for mineral exploitation from the geologist description of a prospect by comparing the observations with models of ore deposits, noting similarities, differences, and missing information. The knowledge about different classes of ore deposits is organised as models. A model contains rules combined with semantic nets. Each rule links any logical combination of pieces of evidence of particular geological findings with hypotheses. The rules form a large inference net, which indicate all the connections between evidence and hypotheses and hence all the possible inference chains that could be generated from the rules. PROSPECTOR has been successful in identifying a mineral deposit previously overlooked.

MYCIN (43) diagnoses meningitis and prescribes drug treatment. In operational trials it has been found to be more accurate and sure-footed than individual experts. All the knowledge on infectious diseases is represented in the form of rules which are expressed in a stylised form that simplifies computer interpretation and facilitates their translation into English. MYCIN's main goal is to apply its rules to determine the identity of all suspicious organisms. When it attempts to apply a rule, it queries its database to see whether the needed facts are available. If there is no information, the program can rely on the user's knowledge or it can use rules to infer the answer. MYCIN's strategy in rule selection is goal oriented and its inference method is to reason backward from its initial goal.

DENDRAL (44) analyses mass expectral patterns to suggest the chemical structure of unknown compounds. It employs an efficient variant of generate and test in its problem-solving strategy. Its generator can enumerate every possible organic structure that satisfies the constraints apparent in the data by systematically generating partial molecular structures consistent with the data and then elaborating them. An effective validation of DENDRAL's performance has been made from various contributions to journals, coupled with its acceptance and routine use by chemists.

The above examples are now considered to be the classical expert systems as each one of these programs have reached or surpassed in some aspects the performance of a human trained in the relevant discipline. The most common expert system is the classifying or diagnostic type in which a choice is based on weighing up the advantages and disadvantages among a number of alternatives. The interrelation of calculation and judgement together with the necessity to cycle through the process repeatedely poses a more complex and difficult problem in design than those met in creating many of the existing expert systems from the diagnostic type (43) to the configuration type (41). Thus, an aircraft design expert system is radically different from existing programs being constructive rather than diagnostic in its inference structure. In addition, it is attractive as an example of a modern design model which will enable the formalisation and clarification of design knowledge from the designer making his reasoning explicit. The structure of the resulting program allows it to be used as a consultation and teaching aid and permits the application of the design concepts to other design domains.

It has to be admitted that within design, little progress has been made and as yet there are no general design expert systems. This lack of progress relates to the fact that design is a highly creative activity involving diverse problem-solving techniques and many kinds of knowledge. However, some success has been reported, particularly on the electronics domain (45), other design domains include the preliminary structural design of high-rise buildings (46) based on a generate-and-test paradigm, design of mechanical components (47) where questions of knowledge representation and use of different control strategies have been addressed, and rule based systems for assembly and coating in manufacturing design (48).

1.2.1.7 <u>Robotics</u> - has addressed a wide range of fields (8) from the optimal movement of robot arms to methods of planning a sequence of actions to achieve a set of goals and thus AI has developed several techniques for modelling states of the world, describing the process of change from one world state to another, understanding how to generate plans for action and how to monitor the execution of these plans. Complex robot control problems have forced the development of methods for planning at high levels of abstraction, ignoring details, and then planning at lower levels where detail becomes important.

1.2.1.8 <u>Combinatorial and scheduling problems</u> - AI researchers have worked on methods for solving several types of combinatorial problems. Their efforts have been directed at making the time versus problem size curve grow as slowly as possible even when it must grow exponentially. Several methods have been developed for delaying and moderating the inevitable combinatorial explosion, with emphasis on using knowledge about the problem domain in order to obtain more efficient solution methods.

1.2.2 Knowledge representation techniques

A representation is a set of syntactic and semantic conventions that make it possible to describe things (9). The syntactic part of a representation specifies the symbols that may be used and the way those symbols may be arranged. The semantics specifies how meaning is embodied in the symbols and the symbol arrangements allowed by the syntax. Much of AI research to date has been concerned with representation of knowledge (8, 9, 49) about the world in a way that can be efficiently collected, stored, and utilized by a computer.

In a knowledge representation system, three levels can be distinguished: the knowledge expressed in a particular formalism, a set of applicable inference rules allowing for the manipulation and derivation of not explicitly stated knowledge, and a control component describing the way in which the inference rules are to be used. The way a computer program represents knowledge affects how it can apply and manipulate that knowledge. A good representation should be explicit about the right things, constraint exposing, complete, concise, transparent, computationaly efficient, detail suppressing, and computable. Experience has shown that designing a good representation is often the key to turning hard problems into simple ones. The most important consideration in examining and comparing knowledge representation schemes is the eventual use of the knowledge, which involves three stages: acquiring more knowledge, retrieving facts from the knowledge base relevant to the problem at hand, and reasoning about these facts in search for a solution.

While the choice of a uniform representation has allowed the construction of large systems, current research is showing a trend towards more complex and heterogeneous approaches. It has frequently been noted that humans exploit several different representations of the same phenomena. For example, in the development of expert systems, experts seem to exploit rule like associations to solve problems quickly but can shift to using more reasoned arguments based on first principles when need arises. This section surveys the representation schemes used extensively in AI programs (8, 49, 50).

1.2.2.1 Logic - has traditionally provided a firm conceptual framework for representing knowledge as it can formally deal with the notion of logical sequence. First-order logic has two aspects: its syntax and its semantics. The syntactic aspect is concerned with well-formed formulae admitted by the grammar of the language. The semantics is concerned with the meanings attached to the symbols in the well-formed formulae. A predicate has a value of true or false and is used to represent relations. A proposition is an expression in which the predicate affirms or denies something about the subject. A constant represents an object in the domain.

An interpretation of a proposition is its truth for a given set of bindings for its variables. Bindings are substitutions of constants for variables. A proposition is valid if it is true for all interpretations. A proposition logically follows from the conditions if it is true whenever its conditions are true. For example,

IF A and B THEN C

C logically follows from the propositions A and B. The introduction of the resolution principle (51) i.e., a single inference rule allows for all possible deductions in first-order logic, created great expectations towards automatic construction of proof but the systems developed were inefficient.

Logic programming (52, 53) is based on two disjoint components: the logic statement of what the problem is that has to be solved and the control statement of how it is to be solved. The ideal of logic programming is that the programmer should only have to specify the logic component of an algorithm. The control should be exercised solely by the logic programming system. Unfortunately, this ideal has not yet been achieved with current logic programming systems. Progress has been made to enlarge the power of logic programming through the enhancement of the expressive power, development of more flexible control mechanism etc. PROLOG is the best known logic programming language.

PROLOG (PROGramming in LOGic) has the mechanisms required for the development of advanced information processing; flexible pattern matching, general data structures, and a search strategy based on backtracking. Its dual semantics (declarative vs procedural) allows the representation of concepts and relationships between objects with great clarity. The PROLOG language has been initially developed and used for AI applications such as expert systems, interaction with the data and knowledge bases, natural language processing, design, decision or planning problems, and other areas where simulation of human intelligence is required. The structure of PROLOG is significantly different from conventional programming languages. A conventional program consists of an ordered series of machine instructions to solve a particular problem which regards data and instructions as totally separate entities and concepts. A PROLOG program consists of a set of facts and rules which describe the application in a very concise, declarative, and clear way. PROLOG removes the distinction between coded instructions and data in order to treat rules and facts sometimes as data to be reasoned about and sometimes as code to be executed. A line of PROLOG code defines something that the computer should know rather than something it should do.

1.2.2.2 <u>Semantic nets</u> - are considered to be the most general and perhaps the oldest knowledge representation scheme in AI. Abstract relationships between objects are commonly drawn as nodes and links. Objects represented by nodes can be physical objects, events, acts, abstract categories, or descriptors. Links represent relationships between objects of the type is-a, has-a, caused-by, etc. For example,

$$\begin{array}{c} \text{aeroplane} \\ \Big\downarrow \text{has\_part} \\ \text{wings} \end{array}$$

Where aeroplane and wings are nodes representing concepts and has_part is the name of the link specifying their relationship. Hierarchical relationships can be constructed and nodes can inherit properties of higher-order nodes. Computational problems arise as the network databases become large and due to the semantics of the network structure e.g., it is difficult to handle exceptions.

1.2.2.3 Procedural representation - The distinction between declarative and procedural representation of knowledge has had a key role in the development of AI. Declarative representations stress the static aspects of the knowledge (facts, events, and their relation) while procedural representations focus on the dynamic aspects of knowledge (how to use the knowledge). Procedural representation systems (8) incorporate ways for explicitly expressing control information in order to direct their problem-solving activity. Thus, partial solutions to problems which are not defined on the entire problem space can be admitted. As a consequence, heuristic information which typically is based on incomplete knowledge can be easily treated. In addition, procedural systems allow for specifying which knowledge should be used to solve a given problem.

1.2.2.4 Production systems - represent knowledge as a set of condition-action rules or productions (8, 54) i.e., 'If this condition occurs, then do this action'. For example,

IF wing mounted engines THEN bending relief

Production rules are the most frequently used method by experts to explain how they solve problems and many well-known expert systems have been built using productions.

A production system consists of a knowledge base where the productions and the data are stored and an interpreter which controls the system's activity by executing rules in some prespecified order until one is found whose condition matches the database, the body of that rule is executed and matching of other rules continues. This account is an idealization of production systems and most of them vary in the form of rules and in the order in which they are executed.

1.2.2.5 Frames - (8) are complex data structures for representing knowledge about the objects and events typical to specific situations in a way that directs attention and facilitates recall and inference. Frames describe values of attributes in a slot or filler format. Each slot can contain a value for the attribute, a procedure for calculating the value, or one or more rules for finding the value. Attached to each frame are different kinds of information such as how to use it. For example,

> When a robot enters a room, it activates a room frame which loads into the working memory a number of expectations about what might be seen next. Suppose the robot perceives a rectangular form. This form in the context of a room might suggest a window. The window frame can then be used to test the confidence in this hypotheses.

Frames are organised in a hierarchical way, with the highest level frame containing information that applies to all frames below it. Any frame related to a higher order frame is said to inherit the characteristics of the latter.

Object-Oriented programming does not represent a true frame language but many of the concepts are similar i.e., an Object-Oriented system has a single type of entity, the object, that represents both the procedures and the data. Smalltalk (55, 56, 57) is the oldest and best known of the Object-Oriented programming languages (58). A Smalltalk program is a collection of objects that communicate by sending and receiving messages. Each object can store some data and has a set of methods (equivalent to procedures, subroutines or functions of conventional languages) which define the messages it can understand. Similar objects are grouped into classes. All the objects belonging to a particular class share the same structure and set of methods. The Smalltalk classes are organized into a tree structure. At the root of the tree is class 'Object', which defines the default structure and behaviour of all objects. All the other classes are subclasses of Object. A subclass inherits its structure and the methods it understands from the parent or superclass, but can alter the structure, and add, delete, or change the set of methods an object (instance) of its class will understand.

1.3 PRESENTATION OF RESEARCH WORK

The research work outlined in section 1.1 and to be described in the following Chapters has been directed at providing an expert system which will assist designers in the preliminary design of civil airliners. The status of the work is one where a prototype program for wing design exists with supporting documentation and a very effective examination and evaluation of the aircraft design knowledge base has been performed using two conceptually different design steps of increasing complexity. The developments and limitations of various concepts and techniques which have evolved for inferencing and knowledge representation are addressed in detail in subsequent Chapters.

Chapter 2 is directed towards the study of the aircraft design knowledge by identifying the sources of knowledge used in the design, classifying this knowledge into different types, and by describing some knowledge representation issues. In Chapter 3 the full aircraft design process is described in all its complex nature with the wing design and aircraft configuration receiving a more detailed description.

A full implementation of the aircraft design process in a single pass is not guarantee to succeed thus, a two pass approach has been adopted as described above. Chapters 4 deals with the first pass by describing the knowledge used by the expert designer in solving the wing design problem. A framework for an aircraft design expert system is described together with its status and illustrations of the knowledge representation features used within the wing design program.

Chapter 5 describes the main details of the resulting output during the second pass in which aircraft configuration is considered. The types of knowledge found in aircraft design identified during Chapter 3 are used to aid in identifying the knowledge representation requirements for aircraft configuration. Extensive examples drawn from the aircraft configuration process are presented for each type of knowledge. Based on this study, the conceptual differences between aircraft configuration and wing design are presented together with the necessary extensions to the aircraft design expert system framework.

Chapter 6 describes a computer program for wing design using the knowledge representation techniques presented in Chapter 4 by describing the scope, procedures used, operating instructions, and a consultation session. The problems with this implementation and the lessons for an aircraft configuration program are presented in Chapter 7. The work is concluded in Chapter 8, with a statement on the status of the current work and recommendations for further developments.

CHAPTER 2

AIRCRAFT DESIGN KNOWLEDGE

2.1 INTRODUCTION

Design (59) is the conception of a component, system, or process to accomplish a specified task optimally subject to certain solution constraints. The design process is not a straight step by step path but, a series of trials and errors. Any one solution may require a number of iterations through various parts of the process and the problem is seldom so simple or the designer so lucky that the first solution is optimal.

Any design can be viewed as a two step process. First, the designer must conceive a method, scheme, or idea that he thinks can work. This is a creative, searching, innovative step. It requires breadth of knowledge and experience, the relating of diverse elements, and an open-ended type of thinking which diverges to and ranges among many possible solutions. The second step consists of analysing the method, scheme, or idea quantitatively to ensure that it can be made to work satisfactorily, subject to given constraints. This requires depth of specialised knowledge, the recognising and remembering of specific facts, mathematical skill, and thinking which converges to a single solution. The two steps are quite different and people skilled at one may not be generally skilled at the other.

The sources of knowledge which help the aircraft designer do a successful job are presented in this Chapter. This knowledge is then classified into different types and various knowledge representation aspects of the design process are described.

2.2 SOURCES OF KNOWLEDGE

The qualities which make a designer successful are the attributes that all successful people have in common, such as perseverance, pursuit of excellence, a pleasant and interesting personality, imagination etc. There is also an element of luck, and a sense of history and an understanding of future trends none of which can be taught. However, there are skills and knowledge which are more closely identified with the designer's engineering education and which can and are taught. The sources of knowledge as shown in Figure 2.1 are described below.

2.2.1 Subjective

The designers in depth knowledge and training in aircraft design together with the associated analytical and computational methods is important. The ability to deal competently and confidently with basic problems or ideas from other disciplines is of particular importance in aircraft design as the designer might be a 'structures man' by training but he must be able to solve problems in aerodynamics, stability and control etc. Futhermore, because of the dynamic nature of the field the designer should have up-to-date knowledge.

2.2.2 Specification

Aircraft design is based on the investigation into and interpretation of the aircraft function and a translation of the most pertinent requirements into a description of more specific tasks. The aircraft must satisfy the performance figures laid down in the design specification and within these limits it must achieve the best economic yield and operational flexibility. The designer should have a clear understanding of the operational requirements of the aircraft such as special requirements regarding visibility from the cockpit, the desirability of the aircraft carrying a very low price-tag and fuel burn etc.

2.2.3 Trend

A specific aircraft is often inspired by a trend or line of evolution which has its origin somewhere in the past. This evolutionary process is influenced by the experience of the design community, competition, the innovations from scientific advances, and individual designer flair.

2.2.3.1 Experience - Good problem-solving requires practice. Designers acquire various sorts of knowledge when they solve or fail to solve a design problem, and when doing one or more rough designs. What they learn in the process enables them to produce a better design next time round. Although the general design requirements will provide important pointers, there is no clear cut design procedure which can be followed and much of the success of the aircraft to be built is based on the experience and resources of the designer.

Factors which help the designer do a successful job include:

1. The ability to analyse a given component, system, or process using engineering or scientific principles in order to arrive at meaningful answers in reasonable time.

2. The ability to make decisions in the face of uncertainty but with a full and balanced grasp of economic factors, technical practicalities, scientific necessities, human and social considerations etc. Thus, the designer should develop a 'design philosophy', determining priorities, indicating solutions etc.

2.2.3.2 Competition and scientific advances - Competition forces manufacturers to explore new and useful ideas which have the long-term effect of advancing the technology. However a major restraining factor is the need to meet existing or anticipated airworthiness requirements. Also, excessively large departures from the existing state of the art may, however, lead to the taking of unwarranted technical and commercial risks. For the conservative operator, the technological advanced aircraft must be easy to handle and maintain.

The development of a new design is very expensive and must be superior to existing designs to justify the required investment. For example, as a replacement for the Boeing 727, Airbus has developed the A320 which offers an improvement on the former aircraft. Boeing on the other hand, has introduced the 757 which offers an 'updated' 727 (e.g., new engines and wings), and has on the drawing board a revolutionary prop-fan design for the 1990's. In other cases (e.g., DC10, TriStar) production has been haulted due to commercial pressures.

2.2.3.3 Flair - refers to the required aesthetics in order to have an aircraft which appeals to operators, passengers and crew. The designer has to have the 'knack' of getting a right compromise between conflicting requirements. This is not only linked to experience, but goes beyond it. For example, a man with 20 years experience might produce an inferior design to that of a less experienced man with flair. The expression 'if it looks right, it is right' is not as ridiculously as it seems.

2.2.4 Environment

The environment relates to how the aircraft is going to be loaded, the airport facilities, atmospheric conditions, how it will be maintained and flown? etc. This represents a combination of factors, some of which can be directly quantified and others which are indirect, require the use of experience or heuristics to quantify.

2.2.5 Production and analysis facilities

Production and manufacturing techniques are dependent upon history and the preference of local management and design teams. For example, some factories traditionally bond components together because they have invested in bonding equipment, know how to design bonded components efficiently and can guarantee a predictable performance. New techniques are introduced into the production very carefully with due consideration given to potential economics. As an example;

robotics are being increasingly used, but they are only justified if the production numbers warrant the investment in new machines. The designer must have the knowledge of and appreciation for the potential of both old and new production and analysis facilities. If his factory does not have the required equipment he can ask for it to be obtained or sub-contract the work. Sophisticated computing and analysis techniques (and suitable staff!) determine the confidence of calculations. If such techniques are not available, conservative assumptions or more testing will be required.

2.3 TYPES OF KNOWLEDGE

Abstractly speaking, knowledge (8, 13) consists of descriptions, relationships, and procedures in some domain of interest. The type of knowledge found in aircraft design is firm, fixed, and formalised (e.g., airworthiness requirements), and at the same time it is subjective, ill-codified, and partly judgemental (e.g., weighing the consequences of selecting a turbofan over a turboprop). In order to represent facts and relationships acquired from the designer in a computer program, the aircraft design knowledge has been separated into the following five types.

2.3.1 Commonsense

Commonsense is used to draw conclusions from partial information and make assumptions as a resource limiting process by default or subject to certain conditions. Also, commonsense is required to draw new information by allowing tentative solutions to be explored when the designer knows what is generally wanted but not how to create it.

2.3.2 Constraints

A constraint is a relation between a set of qualitative or quantitative descriptions. There are two kinds of constraints in the design problem. One set of constraints applies to the designer's problem-solving procedure and consists of such items as his own knowledge, time, experimental and computational facilities available. The other set of constraints applies to the problem solution and consists of such items as cost, nature, availability of materials, equipment, or manufacture skills. The status of a constraint may be satisfied, unsatisfied, withdrawn, or proposed.

2.3.3 Subproblem interaction

Abstraction emphasises the important considerations of a problem and enables its partitioning into subproblems. Designers factor the design into subproblems in order to cope with the complexity of the task. Subproblems are said to be nearly independent when they can be solved with little coordination of the solution process. Typically these subproblems are partially constrained, meaning that they permit more than one solution when only local constraints are considered. Thus, the designer needs to consider not only the local constraints but also the global constraints in order to cope with subproblem interaction.

2.3.4 Metaknowledge

Metaknowledge refers to knowledge about knowledge. It is required in order to:

1. Give priorities to conflicting goals, achieving the important ones first. For example, limiting the time and cost of the design process is often an important goal.

2. Satisfy those goals that are part of several major goals before other subgoals.

3. Check preconditions before executing an operation.

4. Use heuristics to guide the search process and to reduce the amount of computation. For example, designers select one design method over another based on such factors as the ease of applying it, likelihood that it will not lead to a dead end, and its predicted impact on the eventual design.

2.3.5 Consequences

Consequences refer to the properties which are inherited when a goal succeeds.

2.4 ASPECTS OF THE DESIGN PROCESS

The sources of knowledge a designer uses during a search for a solution have been classified into five different interrelated types. This section extends the above presentation by identifying various aspects of the design knowledge which have to be taken into account when defining the knowledge representation scheme to be used.

2.4.1 Incomplete knowledge

It cannot be assumed that the knowledge base is a complete description of the world it is intended to model (60). This observation has important consequences for the operations defined over the knowledge base (inference, access, matching) as well as the design methodologies for knowledge bases. Closely connected with the issue of incomplete knowledge is the issue of default reasoning. In every day situations, decisions have to be made in absence of explicit information about certain facts. In these cases, general knowledge can be very useful for inferring reasonable conclusions.

2.4.2 Approximate and plausible reasoning

In cases where incomplete knowledge exists the system may be able to determine the choice at some stage of the problem-solving process thus guessing is needed in order to continue with the solution. For example, because of the size of the task a designer cannot immediately assess the consequences of design decisions thus, guessing can be an efficient way to explore design possibilities tentatively. Guessing can also be used to obtain a rapid solution to a convergent series of solutions. The difficulty with guessing is in identifying wrong guesses and recovering from them efficiently.

2.4.3 Unreliable data

Experts sometimes make judgements in a hurry, under the pressure of a deadline. All the data may not be available, some may be suspect, and some of the knowledge for interpreting the data may be unreliable. There exists an obvious necessity for recognition and proper handling of such inconsistencies with the formalization of extra metaknowledge in order to correct the data, take back assumptions, or combine evidence.

2.4.4 Spacial knowledge

Aircraft design involves the manipulation of physical objects. Thus, it is necessary to find suitable representation and reasoning methods to deal with spatial knowledge.

2.4.5 Different points of view

In aircraft design there is a need to take into account the knowledge acquired by different means or to be used with different objectives.

2.5 CONCLUSIONS

Knowledge representation is fundamental to the development of expert systems. The present Chapter has been concerned with identifying the sources of knowledge in aircraft design for the purpose of classifying this knowledge together with describing various aspects of the design knowledge. The following Chapter describes the aircraft design process in its full complexity with special reference to the wing design and the aircraft configuration problems.

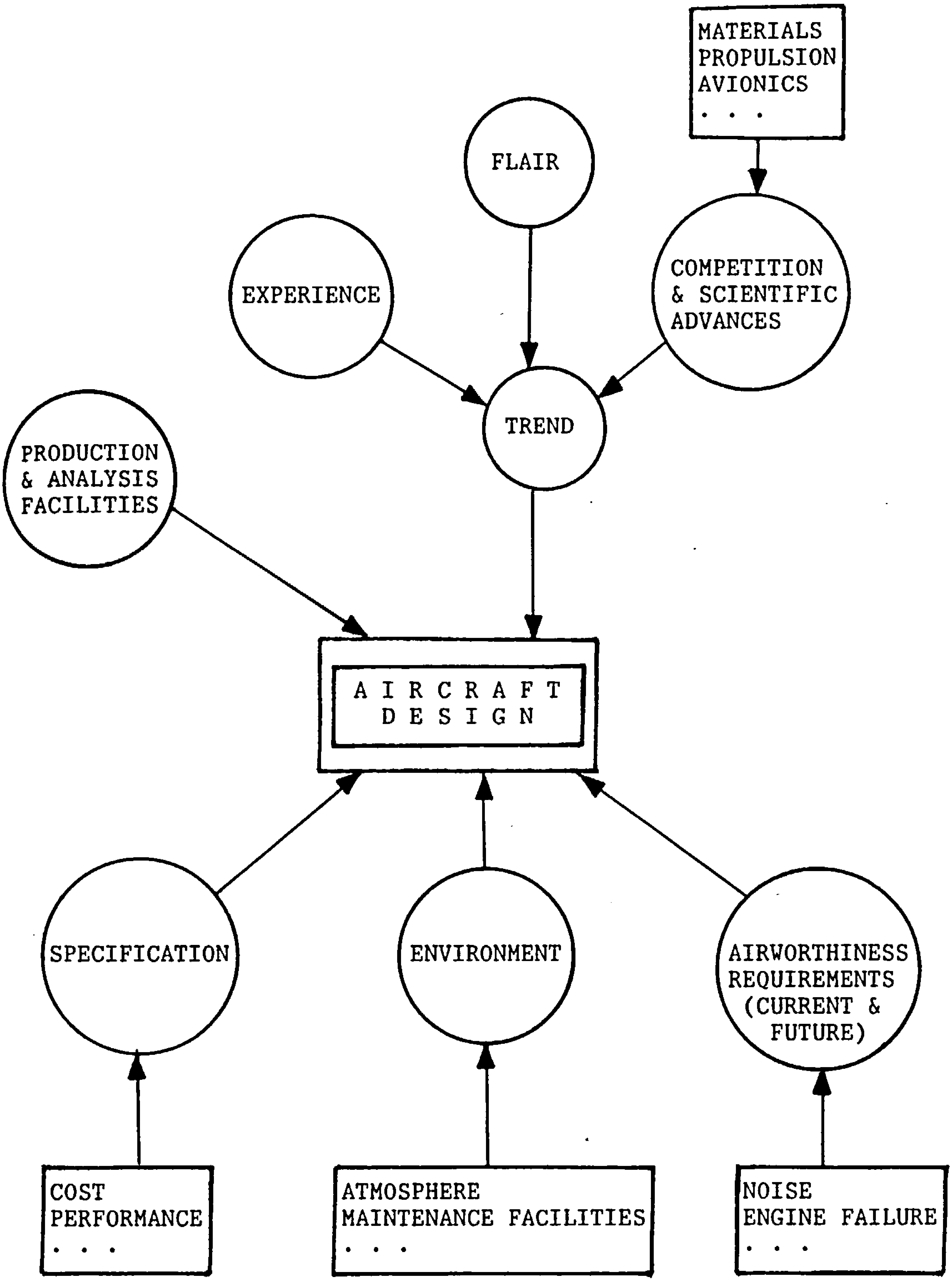


FIGURE 2.1 KNOWLEDGE SOURCES IN AIRCRAFT DESIGN

CHAPTER 3

THE AIRCRAFT DESIGN PROCESS

3.1 INTRODUCTION

The sources of knowledge described in the previous Chapter form the basis of the aircraft design process where knowledge is structured and reasoned upon to create the aircraft itself. In practice, the actual knowledge employed in aircraft design (1, 2, 3, 4, 5, 6) involves the application of the fundamentals of aerodynamics, materials and structures, propulsion, flight mechanics, operational analysis, statistics and optimization and also requires a certain degree of judgement and experience of the designer. It is this 'library' of information which enables the designer to make his first guesses and so get started. The aircraft must satisfy the performance figures laid down in the design specification and within these limits it must achieve the best economic yield and operational flexibility. Essentially the design process is one of successive approximations with the incorporation of fresh data and new requirements as the design proceeds.

The aim of this Chapter is to outline the first few design steps as implemented at Cranfield (1) namely, specification, data bank, parametric study, fuselage design, wing design and aircraft configuration. The scope of the present work, however, has been limited to the wing design and the aircraft configuration as it is felt that these two steps are characteristic of the problem-solving techniques employed during aircraft design as a whole thus, a more detailed description of these two steps is provided. The remaining part of the design process as shown in Figure 3.1 consists of highly analytical steps which are primarily used to examine the various configurations under consideration in line with the specification.

3.2 SPECIFICATION

The design process starts with a specification for civil aircraft which is put forward by both the manufacturers and operators (61). The former highlighting the benefits of new technology and concepts with the hope of stimulating the markets to demand new aircraft. The latter recognising the limitations or outmodedness of its current fleet and putting forward a new specification based on estimates of its future requirements using projections of future traffic and route patterns. Typical items from the specification are (1, 61):

1. The aircraft must have the potential of making a profit for the airline. This is reflected in targets for direct operating costs, return on investment, and break-even load factors.

2. Satisfactory flight characteristics both at high and low speeds and at high and low altitudes for various configurations.

3. The design structure must satisfy demands regarding strength, rigidity, weight, service life, accessibility, development and manufacturing costs.

4. Definition of airworthiness, reliability and safety standards.

5. Other requirements include: fuel consumption, passenger acceptance, noise level, time scale, number of expected production run etc.

3.3 DATA BANK

In view of the current 15 year development period (7) required for a modern aircraft, a new type will only be successful if it is better than the old designs which it is intended to replace and preferably better than old designs competing for the same slice of the market. New designs often develop from previous designs in an evolutionary process using experience built up within the design team. Additional controls on design variations are imposed by the airworthiness authorities which lay down standards and safety requirements. It is true to say today as in 1949 (62) that

> Modern aircraft have become so complicated that the stage between the inception of a new design and the final acceptance trials has become both lengthy and expensive.

Therefore, large departures from existing state of the art can lead to unwarranted commercial risks and may lead to difficulties in airworthiness certificates. The term data bank is used to denote the experience built up within the design team from aircraft in a similar category. It contains data relating to the old designs being replaced and to new competing designs together with requirements laid down by the airworthiness authorities.

3.4 PARAMETRIC STUDY

A parametric study is a design investigation based on general calculation methods and sizing procedures, making it possible to vary configuration parameters and quantify their effect on the design. It consists in using simple empirical expressions (2) to optimize variables such as maximum lift coefficient, aspect ratio, wing loading, thrust loading, whilst satisfying the requirements of take-off and landing performance and economic cruise in line with the specification.

3.5 FUSELAGE DESIGN

The fuselage fulfils several functions, but primarily serves as the load carrying part of the aircraft. Structurally it must absorb the loads (shear forces, torsion and bending moments) transmitted from the wing, tail, power plants and undercarriage. Aerodynamically it must have low drag characteristics and good interaction with other aircraft components (e.g., wing, tail, power plants). In designing a fuselage the following structural and layout considerations should be taken into account (1):

1. Simple structural design giving an efficient and flexible layout with little loss of space, and the possibility of increasing the length of the fuselage if necessary.

2. The cabin should be unobstructed with easy access to the aircraft services so as to reduce the flight turn around time.

3. The number of passengers determines the number of emergency exits, toilets etc. At least one window close to each passenger must be provided.

4. There should be sufficient levels of passenger comfort. This is expressed by seat widths, aisle widths, seat pitches, adequate toilets and galley facilities.

5. The fuselage should have a good aerodynamic shape and provide adequate crew vision and tail clearance for take-off and landing.

3.6 WING DESIGN

Wing design (1, 2) is an important part of aircraft design on which range, maximum speed, manoeuvrability etc., are largely dependent. Its design will force the designer to consider the aircraft as a whole and hence fix some of the assumptions. Aerodynamically, a wing must satisfy requirements for lift, drag, flying qualities, drag rise, aeroelastics, flap effectiveness, and tip stall. Structurally, a wing should have adequate strength and stiffness in bending and torsion, transmit the aerodynamic and inertia loads, possess long life characteristics etc. Consideration should also be given to the use of composite materials and 'active controls' (63) to reduce the wing bending moments and hence the effective load factor.

The aircraft specification will give the required Mach number and the up-stream parametric study will have determined the aspect ratio, maximum lift coefficient at take-off and landing, optimum cruise altitude and lift coefficient. With this information the wing design process can continue. In designing high performance aircraft emphasis is placed on speed as a major factor contributing to the economy and the operational suitability of the conceptual design. The basic considerations for attaining a high cruise Mach number are the adoption of sweepback (or sweep forward), design of improved aerofoil sections and optimum distribution of spanwise twist, camber and taper. Figure 3.2 shows the wing design steps described below.

3.6.1 1st STAGE: two-dimensional wing design

The first decision to be made is to select an aerofoil shape which because of the desire to create an efficient design, will probably lead to the use of a supercritical section. These sections have the characteristic of generating lift right across the chord instead of concentrating lift at a point immediately behind the leading edge as on conventional sections thus, delaying the effect of drag rise, see below.

3.6.1.1 Stall characteristics - The primary function of the wing is to provide lift which a designer can affect by varying the wing section shape (lift coefficient) and the wing area, and a pilot by varying the cruising altitude (air density) and the air speed. At high angles of incidence and low speeds, the stalling characteristics of an aircraft determine the handling qualities and the ability of the pilot to recover from a complete stall. As the stall is approached the flow over the upper surface of the wing separates with the increase in drag and a sudden decrease in lift. The stalling speed of a wing is affected by the wing loading and maximum lift coefficient while the stalling behaviour is affected by the planform, aerofoil sections and wing twist.

3.6.1.2 Cruise lift/drag ratio - The drag is the force in the same direction as the incoming air flow experienced by the wing, it depends on the shape of the wing (drag coefficient), frontal area, air density and velocity. The drag can be divided into two components: lift-dependent drag and zero-lift drag. The former is a function of the wing lift and can be minimised by making the wing long and slender and by ensuring that the lift is distributed right across the span in an efficient manner. The latter is proportional to the square of the speed and depends on the shape of the wing and the skin friction, here the designer has to compromise between a short and bluff shape with low skin friction drag but high form drag and a long and stream lined design with low form drag but excessive skin friction drag. Thus, the cruise lift/drag ratio is a measure of the efficiency of the wing.

3.6.1.3 Pitching moment - At all speeds the aircraft must be in equilibrium about the centre of gravity with respect to the imposed loads. At the specified cruise Mach number and lift coefficient the pitching moment should be of low to moderate magnitude to prevent a high trim drag and torsional moments required to preserve equilibrium.

3.6.1.4 <u>2-D drag rise</u> - is characterized by a rapid rise in both zero-lift drag and induced drag caused by the compressibility effect (the formation of shock waves) which causes flow separation. The flow velocity at which this effect occurs is called the critical Mach number. At the specified cruise Mach number the critical Mach number should be sufficiently high to avoid compressibility effects.

3.6.2 2nd STAGE: three-dimensional effects

We must achieve an acceptable design in terms of 3-D drag rise, aeroelastics, tip stall, flap effectiveness and weight.

3.6.2.1 <u>3-D drag rise</u> - depends on the aerofoil section, thickness/chord ratio (t/c), sweep angle, aspect ratio, and wing-fuselage interaction. Possible ways of delaying drag rise together with some of the 'side effects' are discussed below (1, 2):

<u>Supercritical wing sections</u> as mentioned above increase the critical Mach number allowing a higher cruising speed.

<u>Smaller t/c</u> increases the critical Mach number allowing a higher cruising speed, but there is a limit to the minimum value of t/c due to structural considerations and handling qualities at low speed.

<u>Wing sweep</u> is a very effective way of delaying and reducing the drag rise effects since it depends on the Mach number of the flow normal to the leading edge. But, there is an increase in landing and take-off speeds, reduction in the maximum lift coefficient and flap

effectiveness, and increase in wing loading towards the wing tip which increases the wing root bending moments.

Spanwise variation of sweepback has been used in certain designs to confer special structural and layout advantages such as enabling inboard local t/c to be increased permitting more fuel to be carried and a place to attach and stow the main undercarriage if required, and a decrease of sweep outboard should improve low speed performance. Unfortunately, in practice it seems to be necessary to have several discrete wing sections, each of which has a constant sweep. The kinks which occur at the intersections are structurally inefficient and cause flow separation. Sweepback is normally used since swept forward wings are subject to divergence and longitudinal instability unless the angle is small.

Reducing the aspect ratio and waisting of the fuselage in the region of the wing delays the drag rise effects.

3.6.2.2 Aeroelastic effects - Recent trends in aircraft design have led to very slender wings in order to reduce weight and it is an effective way to achieve long range and high cruising altitude. As consequence, distortion ' or 'aeroelastic' effects have become important. These include:

1. Flutter which involves the interaction of aerodynamic, elastic and inertia forces to produce unstable oscillations.

2. Control effectiveness and reversal i.e., when an aileron is deflected downwards the additional lift created tends to 'untwist' the wing reducing the effectiveness of the control,

at the 'reversal' speed the untwisting effect can be so great that the aileron produces the opposite effect to that desired.

3. Divergence where at a certain 'divergence' speed the aerodynamic moment is greater than the elastic restoring torque and the wing diverges and can break off.

4. Dynamic response to loads and accelerations developed during manoeuvres, heavy landings, taxing etc which are often more severe due to 'dynamic overshoot' effect from the inertia forces.

These snags can be prevented by careful analysis and by such measures as shifting the aeroelastic axis, twisting the wing, repositioning the power plants, using high speed ailerons and spoilers, and using carbon fibre construction.

3.6.2.3 Tip stall - denotes the stall which starts near the wing tip as the wing is brought to a high incidence. A highly tapered wing reduces pitch-up tendency and has a high torsional rigidity but, if it is tapered sharply there is a notable reduction in the maximum lift coefficient near the tip thus aggravating the tendency towards early tip stall. Twisting the wing (wash-out) will reduce the tip stall effect by changing the lift distribution across the wing but, it reduces the lift/drag ratio. The introduction of camber reduces the tip stall problem by changing the profile drag of the section.

3.6.2.4 Flap effectiveness - Increasing the aspect ratio, high angles of incidence may be required in order to generate enough lift and reduce the approach speed making the judgement of the landing point more difficult for the pilot and gives the aircraft a tendency to 'float' after the landing flare. In order to overcome these difficulties an effective flap system is required. Flaps and other devices change the wing camber and planform area enabling relatively high lift coefficients to be achieved at low speeds where the drag is not necessarily so critical, and low drag at high speeds where lift coefficient is of less importance. Leading edge flaps and slats are often used in conjunction with high speed wing sections to prevent premature flow break away.

3.6.2.5 Wing weight estimate - Fuel consumption is proportional to aircraft weight of which the wing typically accounts for one tenth of the empty mass. If we want to fly a greater distance we could increase the fuel capacity - but there is a law of diminishing returns. It is in fact much better to reduce the aircraft weight including wing weight as much as possible by optimising the main load carrying elements and paying careful attention to detailed design. Also, consideration should be given to the use of composite materials and active controls in the design.

3.6.3 3rd STAGE: wing planform

Wing planform refers to the general shape of the wing as viewed from above and takes into account such factors as sweep angle and aspect ratio. A comparison with similar aircraft allows the final planform to be selected. This fixes the position of the kink and the wing tip chord and makes allowances for major structural components, fuel storage, undercarriage and engine location.

3.6.4 4th STAGE: design for take-off and landing

The high lift devices i.e., trailing edge flaps, lift dumper etc., work towards the maximum lift coefficient values produced by the parametric study. At this stage it is necessary to take into account the proportions of the wing span used by the fuselage and ailerons. The take-off and landing performance must be checked when the aircraft configuration, aerodynamics, and weight are well defined. There may then be further modifications to the flaps.

3.6.5 5th STAGE: final sizing

The parametric study gives the optimum wing loading thus, when the total aircraft weight is known, the required wing size may be determined.

3.6.6 6th STAGE: iteration

However, because the weight of the wing contributes to the loads to be carried by the wing the design will not be correct at first pass through the sizing process. Thus, the above stages are repeated with an increasing accurate estimate of the wing size and weight until the process converges to a solution.

3.7 CONFIGURATION

Aircraft configuration (1, 2, 3) is described in this section in terms of the advantages and disadvantages of different configuration choices for each aircraft main component as shown in Figure 3.3. It is found that the arrival at a suitable solution cannot be laid down in a universal, detailed procedure but rather that it is made of one or more iterations. Typically, after a trial configuration has been subjected to a first analysis of its characteristics (weights, mass distribution, performance, flying qualities, economy etc.), it will be seen either that it does not meet all the requirements, or that improvements in some respects are possible.

3.7.1 Engine configuration

Engine configuration (1, 2, 64) as illustrated in Figure 3.4 consists of selecting the best compromise between engine type, size, and position. In performing engine configuration two assumptions have been made, one is that the total thrust (power) to be supplied by the engines is approximately known (from the parametric study) and the other is that the thrust reversal system and the Auxiliary Power Unit (APU) will not be designed but its presence should be taken into account at an early stage.

3.7.1.1 Engine type - Comparison between the different types of engines is difficult and can be misleading. This is because of the many varying parameters which must be considered such as flight speed and duration, altitude, and the structural and aerodynamic configuration of the aircraft. Fortunately the problem has largely solved itself since, with only few exceptions, the role of each type of engine is well defined except in some cases where the aircraft design will be based on an engine project for which certain characteristics are still subject to variation. The following considerations (2) have to be taken into account when selecting an engine type:

1. Limits regarding engine output (rating) and operational conditions i.e., temperature, altitudes, speeds.

2. The thrust and fuel consumption for various engine ratings, altitudes, and airspeeds. (Fuel now accounts for more than 50 percent of the airliner direct operating cost.)

3. Engine weight, dimensions, and location of the centre of gravity (cg) have to be considered when installing the engine.

4. First and maintenance cost.

5. Engine noise and vibration.

6. Passenger appeal e.g., passenger preference for jet propelled aircraft over propeller driven aircraft.

There are two groups of aircraft according to the type of engines used, propeller driven and jet propelled aircraft.

Propeller engines operate at Mach numbers between 0.2 and 0.65. The design of the propeller should fit both the engine characteristics and the performance of the aircraft. The geometry of the propeller is also important in view of the clearance between the propeller and the airframe or the ground. Three types exist:

1. Piston engines no longer have an application in aeronautics apart from small, light aircraft where first cost is of prime importance, and it is unlikely to be used in aircraft flying faster than Mach numbers (M) equal to 0.5. The installation weight for a given power is relatively high. An important

consideration with piston engines is the method of cooling. The vast majority of current engines are air-cooled since it is a very simple system but it does involve a considerable drag penalty, which is not too critical in light aircraft. A more complicated alternative is the liquid cooled engine but it is heavy and the radiator that is used to cool the liquid produces drag. Also, the piston engine burns gasoline rather than cheaper kerosine. The maximum power currently available is 400hp, greater power requires the use of more engines, or turboprops.

2. Turboprop engines are universally used in aircraft flying at Mach numbers less than 0.65 except in the smallest aircraft despide a more complex propeller control mechanism and higher cost per horsepower than for piston engines. It has the advantages over the piston engine in having a lighter weight, lower fuel consumption, smoother running, and a source of compressed air.

3. Propfan driven aircraft could be the next major advance in commercial aircraft technology (65, 66). They differ from older propeller designs because they can operate at jet speeds, but with far greater efficiency. Propfan engines use modern technology jet turbines to power the propeller systems. The propellers differ from earlier designs by using more blades, single- or contra-rotating, which are highly contoured for aerodynamic efficiency and are usually mounted on the aft end of the engine. Mach numbers between 0.7 and 0.8 can be obtained without blade compressibility effects.

On an average short-haul trip, the fuel consumption could be reduced by up to 35 percent that of the most advanced turbofan engines available, and possibly by as much as 50 percent under many aircraft currently in service. But, problems remain regarding structural and ground clearance, and near and far field noise.

Jet propulsion (64) is used at Mach numbers greater than 0.65 due to blade compressibility effects on conventional propeller engines which reduce efficiency, while offering a relatively high degree of freedom when positioning the engines. Two types are observed:

1. Turbojet is used at Mach numbers between 0.75 and 3.0 at altitudes of up to 18Km but, for aerodynamic reasons few aircraft operate between 0.9 and 1.4. The cost and complexity of the turbojet rules out its use in many applications although, some light aircraft use turbojet engines at Mach numbers down to 0.55 where the layout advantage of eliminating large propellers offset the weight increase relative to the turboprop. It is also very noisy and it has much worse fuel consumption than the turbofan.

2. Turbofans or Bypass has the advantage over the turbojet of reducing both fuel consumption and noise and taken to the extreme may use a geared, variable pitch fan which effectively makes the engine a ducted turboprop. Although, they run out of power at very high altitudes and sonic speeds, and are heavier and more complicated than turbojets.

3.7.1.2 Engine size - and number of engines are determined from the thrust/weight ratio performed during the parametric study. In theory the minimum number of engines required is one and this number produces the most efficient design but, in practice the minimum number of engines required is two (67). Some of the considerations in using 2, 3, or 4 engines are listed below.

1. 4 engines need to be less powerful than 3, which in turn need to be less powerful than 2. This is because there is a requirement for safe take-off in the event of the failure of a single engine. Thus, the failure of one in a twin engined aircraft engines would result in a thrust loss of 50 percent whereas in a four engined aircraft is 25 percent.

2. 4 engines weight more than 3 which in turn weight more than 2.

3. 4 engines cost more than 3 which in turn cost more than 2.

4. 4 engines produce more drag than 3 which in turn produce more drag than 2.

5. 4 engines will cause more unscheduled withdrawals due to failure than 3 which in turn will cause more than 2.

6. 4 engines will cost more to maintain than three, which in turn will cost more to maintain than 2.

7. 4 engines carried on the the wing will save more structure weight than 2.

8. 4 engines mounted on the fuselage penalise the weight more than do 3, which in turn are worse than 2.

9. Present regulations state that more than 2 engines are required if long haul over-water routes are to be used.

The length of the take-off runway has proved to be the most useful parameter for comparison purposes (2). For example, it is generally found that two jet engines are used for short-haul aircraft and three or four for medium and long ranges.

3.7.1.3 Engine position - and installation are one of the most difficult problems in project design (1, 2, 3, 68). Factors which should be investigated include:

1. Bending relief and landing impact loads.

2. Engine maintenance.

3. Foreign object ingestion.

4. Fuel, anti-icing, and air conditioning.

5. Passenger and cargo loadability.

6. Far and near field noise levels.

7. Performance in terms of drag, maximum lift, and second segment climb.

8. Flying qualities.

The two main possible layouts are the wing and fuselage mounted engines.

Wing mounted engines present the problem of trying to achieve favourable aerodynamic interference with the wing or, at worst, minimise the unfavourable interference. Wing mounted engines can be overwing, underwing or midwing, yielding one or more of the following consequences:

1. The engines can be easily accessible from the ground.

2. The mass of the engines and pylons lead to a reduction in bending moment at the wing root, thus lightening part of the wing structure.

3. The engines pylons can have a favourable effect on the airflow at large angles of attack and tend to counteract pitch-up of sweepback wings.

4. In case of the aircraft crashing, the engines can absorb some of the impact energy.

5. There is loading flexibility due to the relative position of the wing and the tail arm, setting the cg position at about half the aircraft length without large variations.

6. The short intake and exhaust ducts enable the engines to run under optimal conditions.

7. Separately spaced engines are well placed from the safety point of view in the case of fire but, engines placed too far outboard increase the landing impact loads and a large yawing moment can result from an engine failure.

8. The engines may be restricted in position by the need to provide ground clearance in order to eliminate, as far as possible, the possibilities of stones and other debris being thrown into the air intake particularly with underwing pod or propeller mounted engines.

9. There is an increase in induced drag and in particular with overwing mounted engines (69).

10. Flap fatigue can be a problem due to engine exhaust.

11. Reduces flap span which can lead to more complexity for a given lift coefficient.

Burying the engines within the wing is nowadays not used in transport aircraft. It yields the following consequences:

1. 'Clean' and relatively big wing.

2. The hole through the wing causes structural penalty.

3. The wing can carry less fuel.

4. There is not much flexibility with respect to being able to change the engine type without structural alterations to the wing.

<u>Fuselage mounted engines</u> fitted towards the rear of the fuselage yield the following consequences:

1. A 'clean' wing with the possibility of using full span flaps.

2. A low cargo floor can be obtained.

3. After an engine failure, there is little yawing moment.

4. Minimises the noise perceived by the crew and passengers.

5. The wing has to be set further back to balance the aft engines reducing the loading flexibility due to the greater cg movement which can cause stability and control problems.

6. Large tailplane is required in order to provide sufficient tail moment due to a shorter tail arm with the consequent loss in lift/drag ratio.

7. A local 'beef up' of the rear fuselage structure is required and leads to a loss of useful space in the tail, resulting in added structure weight and a larger fuselage for the same payload.

8. Engines mounted on the rear of the fuselage are often combined with a tailplane on top of the fin.

9. Due to the jet exhaust there can be structural damage (e.g., fatigue) done to the tailplane, fin and rear fuselage.

10. In the event of the aircraft crashing, the inertia forces from the engines can damage the wing fuel lines.

11. Hot air from the engines (used as de-icing) and fuel lines must pass through the fuselage.

Another possible arrangement is that of the three-engine aircraft where one engine is generally mounted centrally at the rear of the fuselage. The problem which will have to be faced here is whether to bury the engines in the fuselage, which will require a fairly long and curved inlet with consequent loss of efficiency and extra weight, or to bury the engine in the tail unit which causes structural problems due to the hole, or have the engine installed in a pod on top of the fuselage, but in that case the vertical surface forms an obstruction. Although the three-engine aircraft adds an extra safety factor to the aircraft operation, the installation of all three engines on the fuselage should be avoided if possible.

3.7.2 Wing configuration

Wing configuration (1, 2, 3) as seen in Figure 3.5 is characterized by the wing design as described in section 3.6 and the horizontal and vertical position of the wing on the fuselage.

3.7.2.1 Horizontal wing position - Positioning of the wing along the fuselage is concerned with the longitudinal stability and control of the aircraft. A very important factor in aircraft configuration is the location of the cg for various loading conditions. The aircraft can be balanced by moving the wing position along the fuselage (fore and aft) in order to obtain a satisfactory cg position. This depends mainly on engine position.

3.7.2.2 Vertical wing position - The wing is normally set at a small angle (1 to 3 degrees) to the longitudinal axis of the fuselage to ensure minimum drag for the whole aircraft and that the fuselage axis or cabin floor will be horizontal during cruising flight. With the interior arrangement, safety, performance and flying qualities, and structural aspects becoming the deciding factors when the choice between high, low and mid wing is not dictated by considerations of maximum operational flexibility. The advantages and disadvantages of each of these positions are listed below.

High wing presents the following advantages:

1. The floor level is low for quick loading and unloading of cargo and personnel, and provides good access to the fuselage for maintenance.

2. There is generally more freedom when positioning the engines relative to the wing particularly with propellers. For underwing mounted pod engines it provides ground clearance and protection against foreign object damage.

3. The pilot and the passengers have a good outside view.

4. Easier conversion into a dedicated cargo aircraft, for example the commuter Shorts 330 converted to the cargo aircraft SHERPA.

5. The wing-fuselage combination has a favourable aerodynamic effect.

The following considerations are put against this layout:

1. Retraction of the main undercarriage poses special problems. In small propeller aircraft it may be possible to retract the main undercarriage into the engine nacelles or in the tail booms, but in the case of larger aircraft in doing so would

make the undercarriage tall and heavy (3). This will lead to mounting the undercarriage under the fuselage which would need to be strengthened to absorb the landing loads increasing the aircraft weight. Also, there is a minimum width requirement between undercarriage legs, this usually means that the legs have to be mounted as far as possible away from the fuselage centre line. With retractable legs, this requires large blisters fairings which add to the aerodynamic drag and weight, and interfere with the airflow over the unswept rear fuselage.

2. Provisions must be made for escape through the cabin roof in case of a forced landing on water.

3. Accessibility to engines and wings for refuelling and maintenance purposes is more difficult.

4. Depending on the fuselage size the wing structure can interrupt the cabin headroom.

5. Damage to the fuselage by stones and other debris can be a problem in rough fields due to the proximity of the fuselage to the ground.

Mid wing is generally chosen when minimum drag in high speed flight is of primary importance. It is generally found in fighter and trainer aircraft provided the space required for the useful load is small in relation to the total internal fuselage volume. For transport aircraft this layout is not a good solution since it presents structural problems and interrupts the cabin.

Low wing is the most familiar layout and offers the following advantages:

1. Good accessibility to engines and wings.

2. Provides a good solution for positioning the main undercarriage with weight savings.

3. The generally greater fuselage height above the ground offers advantages when, after a fuselage stretch, the tail angle available is still sufficiently large to allow for optimum rotation during take-off, without creating unacceptable geometrical pitch angle limitation.

4. The low wing and possibly the engines will form a large energy absorbing mass during a forced landing, although they can also present potential fire hazards upon contact with the ground. If the aircraft is forced down on water, the wing can serve as a floating platform. Emergency exits over the wings ease passenger and crew exit.

5. The ground effect reduces the take-off distance but increases the landing distance.

Disadvantages of a low wing layout:

1. Difficult to ensure enough ground clearance for engines.

2. On large aircraft the high floor makes the aircraft dependent on special loading and boarding equipment.

3. If the aircraft has a fuselage diameter less than 3 metres then it becomes difficult to fit the wing underneath the fuselage, and flow separation at low speeds can be a problem. Fairings at the wing root can be installed to minimise this problem but the aircraft weight goes up.

3.7.3 Undercarriage configuration

The landing gear (1, 2, 3, 4, 70, 71, 72) of modern aircraft (tyres, wheels, brakes, landing legs and associated retraction equipment) represents a substantial unit of the aircraft, it accounts for some 3 to 5 per cent of the maximum take-off weight, which is equivalent to 15-20 per cent of the structural weight of the aircraft. The maintenance costs associated with the undercarriage represent a considerable item in the total maintenance bill. This is particularly hard to accept because the undercarriage contributes virtually nothing to the flying and economic capabilities of the aircraft. The basic design requirements are (71):

1. It must be capable of absorbing a certain amount of energy, both vertically and horizontally. In addition, the undercarriage characteristics must be adapted to the load-carrying capacity of the airfields from which the aircraft is intended to operate.

2. During taxying, take-off and landing no other part of the aircraft will touch the ground.

3. No instabilities must occur, particularly during maximum braking effort, crosswind landings and high-speed taxying.

Figure 3.6 shows undercarriage configuration as described below in terms of selecting a type, position and size.

3.7.3.1 <u>Undercarriage type</u> - Take-off catapults, take-off trolley, snow skids, four castors mounted at the tip of the wings and rudders have been developed for special purpose aircraft. However, most aircraft have three wheeled elements, with either an auxiliary front or rear wheel, most load being carried by a pair of wheels near the cg A few aircraft have been built on the so called bicycle or tandem layout in which the mainwheels are arranged practically in the plane of symmetry of the aircraft and the front and rear wheels absorb landing impact loads of the same magnitude, auxiliary outriggers prevent the aircraft falling over sideways when taxying or at rest, the structure must also withstand the loads imposed on it during an outrigger first landing. The main considerations in selecting a type are: stability and performance during lift-off, touch down, and taxying, retraction and positioning, cost. The types observed in practice i.e., tailwheel, nosewheel, and bicycle are described below.

<u>Tailwheel layout</u> is now obsolete for most designs, its advantages should nevertheless be mentioned:

1. It is usually the lightest for the same aircraft structure.

2. The location of the auxiliary wheel is a relatively unimportant part of the fuselage.

3. Some of the total forward energy is dissipated in air drag due to the tail down attitude when landing, energy which otherwise would have to absorbed by the brakes.

4. During braking the aircraft tends to pitch nose down, increasing the mainwheel reaction and reducing the possibility of skidding.

In spite of these advantages the tailwheel type has fallen from favour, largely because of the following disadvantages:

1. Drag forces due to braking, act forward of the cg and are thus destabilising in yaw so that there is a strong tendency for the aircraft to swing. Heavy braking can cause the nosing over or overturning of the aircraft.

2. In a two point landing the resulting pitching moment is nose-up. The increase of incidence implied by this is likely to cause an increase of lift and the aircraft will bounce.

3. Take-off is made more difficult by the increased drag until the tail can be raised.

4. When taxying the pilot has difficulty in seeing where he is going unless the aircraft is 'swung' from side to side.

5. Loading of the aircraft with cargo and passengers is complicated by the inclined floor line.

Nosewheel (tricycle) layout is considered typical in today's airliner. The major advantages are:

1. Heavy braking cannot cause nosing over or overturning because the nosewheel acts as a prop.

2. Brake drag forces act behind the cg and are therefore stabilising in yaw.

3. The initial take-off attitude has a low drag.

4. The nose down pitch resulting from a two point landing helps to shed lift and prevents bouncing although lift dumping may still be required.

5. The view of the pilot is relatively good.

6. Whilst the aircraft is on the ground, the fuselage, and hence the cabin floor, is always roughly horizontal.

The increase in landing speed of modern aircraft has accentuated these advantages so that they more than outweight the following disadvantages:

1. High weight of the nosewheel.

2. The need for a tail bumper, or locally stiffened rear fuselage.

3. The main and nosewheels may be difficult to mount because of their location to a suitable structure.

4. Any tendency towards an aft movement of the cg is detrimental to ground stability.

5. The horizontal attitude of the aircraft during deceleration reduces the aerodynamic drag which assists the brakes.

6. As the mainwheels are aft of the cg, the application of brakes and implied nose down pitch reduces the reaction forces causing a strong possibility of skidding to arise.

7. The retraction of the nosewheel can be difficult.

<u>Bicycle layout</u> has many disadvantages but one or two particular advantages have resulted in its application in certain designs, such as for bomber and vertical take-off and landing (VTOL) aircraft. The important advantages are:

1. The main load carrying wheels are located roughly equidistant fore and aft of the cg, thereby leaving a substantial length of the aircraft about the cg clear of obstructions.

2. The wheels can be stowed in the fuselage thereby ensuring a good wing structure.

Disadvantages against this layout include:

1. Outriggers are necessary and unless care is taken in the layout they can become large and heavy.

2. The aircraft landing attitude must be carefully controlled.

3. A large elevator power is necessary to raise the nose during take-off.

3.7.3.2 Undercarriage position - The most satisfactory procedure in laying out the undercarriage (71) is to consider first the wheel disposition in elevation for the landing and take-off conditions, and then the arrangement in plan based on conditions of stability during taxying, lift-off and touch down. Some of the questions which need answering include:

1. Where will the main legs be located - fore or aft?, where will they be attached to?, will they obey ground stability criteria?.

2. Where will the main leg be located laterally?, will it obey ground stability rules?.

3. Is there sufficient ground clearance for the rear fuselage, engines, flaps etc., at all expected aircraft attitudes?.

4. If a retractable undercarriage has been chosen, where will the undercarriage be stowed?, is there sufficient room? if not, are blisters acceptable?. Is the retraction direction compatible with free-fall in the event of power failure?.

Disposition of the wheels in elevation require that the rear of the fuselage remains clear during take-off and landing, and that during taxying there should be enough clearance between the ground and any other part of the aircraft. For each undercarriage type the main considerations are described below.

1. Tailwheel layout The tailwheel will be mounted as far aft as possible to reduce the load on it, but compatible with stowage and structure. The mainwheels will be mounted forward of the cg position along a line from it substending an angle with the normal to the ground of about 17 degrees as shown in Figure 3.7. An angle less than 15 degrees can be dangerous when heavy braking is used because of the possibility overturning. The danger of overturning is caused by forces acting sideways on the aircraft such as a crosswind, angle of yaw relative to the runway, high speed turn during taxying, or taxying over uneven surfaces. Therefore allow the angle to be less than 16 degrees only with the cg position in an adverse position, retaining 17 to 18 degrees under normal conditions. On the other hand, a too far forward position will increase the load on tailwheel, and there will be difficulty in obtaining sufficient elevator moment to lift the tail at take-off, to allow the aircraft to accelerate.

2. Nosewheel layout The nosewheel should be mounted as far forward as possible to reduce the load on it, minimise elevator power required for take-off, maximise main gear load for maximum retardation during braking, and maximise stability. A too far forward mounting may cause a greater structure penalty in carrying the loads down to the fuselage structure than would be saved in the weight of the nosewheel itself. The mainwheels will be mounted behind the cg but not too far to keep the nosewheel loads low.

Knowledge of the wing incidence on the approach enables the ground line to be drawn relative to the aircraft datum. There should normally be about 15cm (6in) ground clearance unless a bumper is specifically incorporated. At touch down it is usual to assume that the shock absorber will be fully extended and this includes any bogie trim angle which may be used. The position of the main wheel, or centre of a multiwheel assembly is determined by ensuring that the contact point is at least 4 degrees behind the perpendicular drawn from the ground line to the most adverse landing cg position as illustrated in Figure 3.8. Excessive aft location of the mainwheel may imply an unacceptably large elevator power for lifting the nose at take-off, or large nose down pitch in landing. Unduly far forward location of the mainwheel implies poor ground static stability. The mainwheel location can be fixed provisionally relative to the nosewheel and the airframe. Consideration must be given to the location of suitable attachment structure and stowage space. In many cases the static ground line is parallel to the aircraft floor line or datum, but sometimes a nose up attitude is desirable to assist the take-off. Location of the nosewheel is initially determined by reference to the static load which it will react. This should be 10 per cent of the aircraft all up weight. Less than 8 per cent implies poor steering adhesion and consequently difficult ground manoeuvring, whilst more than 14 per cent results in a heavy unit and ground instability.

3. Bicycle layout The fore and aft location of the wheels is likely to be mainly determined by the structural or bomb door requirements. The optimal layout will give about equal loads under dynamic braking conditions, otherwise the front and rear tyres may not be of equal size. The static loads will be 55-60 per cent on the rear wheels and 40-45 per cent on the front wheels. The outrigger wheels or skids should be reasonably close to the cg in elevation to eliminate any tendency to pitch the aircraft in a fore and aft plane when touching down one outrigger first. They should clear the ground when landing with a small angle of bank but not too much, otherwise as the aircraft taxies, the centrifugal force set up when turning may cause high dynamic loads on the outer auxiliary wheels. A swept back wing will complicate the installation of the outriggers as they may conveniently be installed only some way behind the cg.

Disposition of the wheels in plan is not as critical as that in elevation. It is often determined by secondary requirements such as stowage or structural attachment. It is necessary to avoid ground instability and nosing over during braking and to ensure that there is adequate ground clearance in all likely landing or ground attitudes. The angular roll of an aircraft under a given side force is inversely proportional to the square of the track, and while this can be alleviated by using a stiff or two stage gear, this may be undesirable from the overall shock absortion and cost standpoints. Thus an aircraft with a narrow track becomes liable to lateral 'wallowing' and a landing gear of low stiffness may be undesirable. A very wide track, however, is usually unnecessary, and makes taxying up narrow perimeter tracks difficult.

1. Tailwheel layout The criterion used is the same as that described below for the nosewheel.

2. Nosewheel layout The track of the mainwheels should be as wide as possible to improve stability during taxying. Retraction of the wheels outwards into the wing usually means a narrow track, and attachment to the fuselage certainly will. However, the important factor is not just the track, but its relation to the cg In order to avoid ground instability and nosing over during braking the plan apex angle of the configuration should not be greater than 80 degrees, and the angle between the ground and the line joining the cg to the side of the triangle in a plane parallel to the mainwheel track should not exceed 55 degrees as shown in Figure 3.9. In fact, this is somewhat conservative as it does not take into account the undercarriage stiffness.

3. Bicycle layout A similar criterion of overturning sideways should be taken as for the nosewheel. The disposition of outrigger wheels on a bicycle layout will probably be determined by the available structure. The wider the track the greater the risk of touching down outrigger first in a banked landing. If the track is reduced too much, the dynamic loads on the outrigger, either when turning at high speed, or when rolling over sideways, becomes greater. The usual track is 40 to 50 per cent of the wing span.

3.7.3.3 <u>Undercarriage size</u> - depends on the mass of the aircraft and the type of the airport to be used. It relates to the undercarriage length, number of legs, number of wheels per leg, the size of the wheels and tyres. The main requirements for the undercarriage length are to provide enough ground clearance to other aircraft components, and to keep the fuselage horizontal or slightly tilted nose down when the aircraft is on the ground in a non-tailwheel undercarriage. The preliminary tyre selection can be made by determining how many tyres will be used on each strut. In many cases the answer can be obtained by looking at similar aircraft. From the maximum main gear static load, the static single wheel load can be obtained by dividing the main gear static load by the number of tyres per strut however. The nosewheel tyre size is usually determined either by the three point landing or dynamic braking. The axle travel is best fixed to ensure zero overall pitching during a three point landing. The tyres can now be selected from the tables provided by the manufacture's catalogues.

3.7.4 <u>Tail configuration</u>

The design of the tail surfaces (1, 2, 3, 4) probably depends more on the general arrangement and the detail layout of the aircraft than any other major part. The tail configuration process as shown in Figure 3.10, must meet the following requirements:

1. The aircraft must possess positive directional and lateral static stability and must meet specified standards of longitudinal dynamic stability in both short and long period (phugoid). Similarly, the aircraft must have acceptable lateral and directional characteristics in roll and

oscillation (Dutch roll). Some degree of spiral instability will be acceptable for aircraft fitted with an automatic pilot.

2. Generation of forces for manoeuvering the aircraft i.e., rotation during take-off, control of flight path, flareout during landing and taxying.

3. The vertical tail must not stall as a result of an oscillation after deflection of the rudder or a sudden engine failure in which the aircraft must remain controllable to ensure steady flight.

4. It should be possible to land transport aircraft in crosswinds of up to 55 Km/h (30 Knots).

5. The tail moment arm must be sufficient to restore level flight if the aircraft enters a 'deep stall' i.e., when both the wing and tailplane stall at high angles of incidence.

3.7.4.1 Tail type - Although there are many intermediate solutions, the discussion is based on conventional types i.e., aft tail arrangements with no canards or forward swept wings.

High-tail (T-tail) consists of a single fin with the stabiliser mounted on top of the fin. The advantages of having this arrangement are:

1. The moment arm from the cg is longer because it is usually associated with a swept back fin thus, a smaller tail volume can be used.

2. The tail is relatively easy to manufacture, leaving a 'clean' fuselage.

3. The intersection between the tail and the fuselage is simplified.

4. Minimises the wing downwash effects on the tailplane.

5. Easy to make an all moving tailplane if required.

6. Clear of rear mounted engines exhausts.

7. Offers advantages in performance because of the lower vetted area of the horizontal and vertical tail combination. The horizontal tail tends to end plate the vertical tail, and thereby increase the effectiveness of this surface.

Against this layout, the following points are put forward:

1. The end plate effect causes side loads on the fin due to rudder application or yaw induced rolling moments of the same sign, increasing the torque reacted by the rear of the fuselage. Thus, the size of the fin-tailplane junction and the stiffness of the rear fuselage need to be increased.

2. Very high loads on the top of the fin due to lift forces and rolling moments have to be resisted by a small structural base thus, the tail unit has to be stiffened.

3. Care must be taken to ensure that a nose down pitching moment can be obtained from the tailplane at incidences well above the wing stall in order to avoid a deep stall.

4. Access to the tail and the tailplane actuator is difficult.

5. Clearance difficulties in factories and hangers can arise for large aircraft.

Low-tail consists of a single fin with the stabiliser mounted on the fuselage, it has the following advantages:

1. Forces and moments can be easily reacted by a wider base at the fuselage.

2. A tailplane position well above or below the extended wing chord is desirable to avoid wing downwash.

Arguments raised against this configuration include:

1. The need for the attachment between the fuselage and the stabiliser to be flat makes manufacturing and sealing difficult. This problem is aggravated if the rear of the fuselage is highly swept.

2. Lower aerodynamic efficiency due to interference effects.

Cross-tail consists of a single fin with the stabiliser mounted more or less midway between the fuselage and the top of the fin. This arrangement usually combines the worst parts of the high- and low-tail configurations. However, the structural penalty is not as bad as for the high-tail even though the aerodynamic advantages are less.

Twin vertical fins consists of two fins set a distance apart at either side of the centre-line of the aircraft. If the rolling moment due to rudder deflection of a large fin is considered to be objectionable, a twin fin may be well worth investigating as a means of minimising this effect. Twin fins reduce the overall height and could be beneficial for aircraft storage, attachment to the fuselage is easier than for a T-tail but, usually imply twin rudders, the end plate effect on the tailplane can be difficult to achieve and sufficient rear fuselage clearance can be a problem on take-off and landing.

V-tail consists of two fins set at an angle to either side of the fuselage. It is often adopted in sailplanes, with the object of avoiding damage to the tail when landing on overgrown terrain and in powered aircraft to keep the tail surface clear of the jet efflux without having to resort to a T-tail. But, it has never become popular, mainly because the moving surfaces have to serve both as rudders and as elevators which leads to complication in the control system design.

3.7.4.2 Tailplane position - Once the vertical placing of the wing on the fuselage has been chosen the location of the tailplane (2, 73) follows fairly readily since the major factors are to minimise wing downwash and the engine effects.

Jet efflux affects the tail surfaces due to the change in airflow direction and the jet pumping effect, disminishing the stabiliser effectiveness. It is advisable to have as great a distance as possible between the noise generating regions and the tail surfaces, and a good rule-of-thumb is to avoid placing anything within a cone of 12 degrees apex angle of the jet pipe.

Slipstream effects. The downwash and the local velocity distribution at the tail depend on the engine speed. When the airspeed and the angle of attack are changed the stabiliser moves in a vertical direction relative to the slipstream, which causes variations in the longitudinal stability. This depends partly on the vertical location of the stabiliser. It can be shown that loss of static stability is small with the stabiliser placed very high or low.

Stability and control of the wing is affected by the wing sweep, aspect ratio, and may also be influenced by the airfoil variation, wing twist, boundary layer fences, engine pylons and leading edge high-lift devices. The recovery from a stall is a safety requirement which must be demonstrated during certification. Most tailplanes designed for normal operating conditions will be sufficiently effective to provide stability at high angles of attack. However, care must be taken when positioning the tailplane to avoid 'deep stall', particularly with a T-tail (73).

Recovery from spins in the case of aircraft designed for aerobatics must be possible. This involves the use of the rudder, which must be effective even at very large angles of incidence.

3.7.4.3 Tail size - The calculation of the tail surface areas in the horizontal and vertical plane (1, 2, 3) is complex and the results obtained must be checked against wind tunnel tests.

Vertical tail (fin) provides the directional control and stability and may be sized by one or more of the following conditions:

1. Low speed one-engine out or severe cross-wind conditions during landing and take-off.

2. In flight with one engine inoperative there will be a yawing moment which has to be counteracted mainly by rudder deflection. There will also be a non-symmetrical lift distribution over the wing and this will cause a sidewash at the fin, effectively resulting in an increase in the yawing moment.

3. The require aircraft manoeuvrability may size the vertical tail.

At this point in the design, there is not enough information to size the tail by any of the above criteria. Thus, the vertical tail area is determined by comparing the vertical tail volume coefficients for similar aircraft (2).

Horizontal tail provides longitudinal control and stability and may be sized by one or more of the following criteria:

1. The static longitudinal stability derivative must be negative at all flight speeds.

2. Provide the required aircraft manoeuvrability.

3. It must be powerful enough to rotate the aircraft about the main gear at rotation speed, and to trim the aircraft at low speed and at the maximum lift coefficient.

4. The drag due to trim load on the tail at cruise must be less than 10 per cent of the total drag.

The horizontal tail area is evaluated by comparing horizontal tail volume coefficients for similar aircraft (2).

3.7.5 Example configuration

The order in which aircraft configuration is carried out does not follow any specific procedure. The relative importance weighting the different factors in a configuration depend on the type of aircraft under consideration and on the designer's experience and prejudices. Figure 3.11 summarises in a 'matrix' form the considerations which have to be taken into account according to the aircraft type with guidelines in configuring an over-water executive jet as described below.

1. Wing layout. A low wing is to be selected giving good ditching over water crash capabilities and the required aesthetics. The wing will be swept back to enable the aircraft to cruise at high subsonic speeds.

2. Engine layout. Choose 2 3 fuselage mounted engines to give extra confidence over water, ground clearance, and minimise the noise perceived by the executives. Centre of gravity movement is not a great problem with this type of aircraft.

3. Tail layout. Choose a high-tailplane to provide engine clearance and relatively long tail arm.

4. Undercarriage layout. Choose a tricycle undercarriage for the same reasons given in section 3.7.3. The undercarriage consists of single main legs, each with 1 or 2 wheels depending on storage.

3.8 CONCLUSIONS

Aircraft design is a large and complex task involving several disciplines related to aeronautical engineering together with a considerable amount of judgement exercise by the designer. In the above sections the first few design steps have been described with emphasis on wing design and aircraft configuration as these two steps highlight the interrelation between the analytical and the judgemental knowledge used in aircraft design. The following two Chapters consider the nature of the knowledge used in wing design and aircraft configuration respetively, in order to meet the goals and objectives of the current research program.

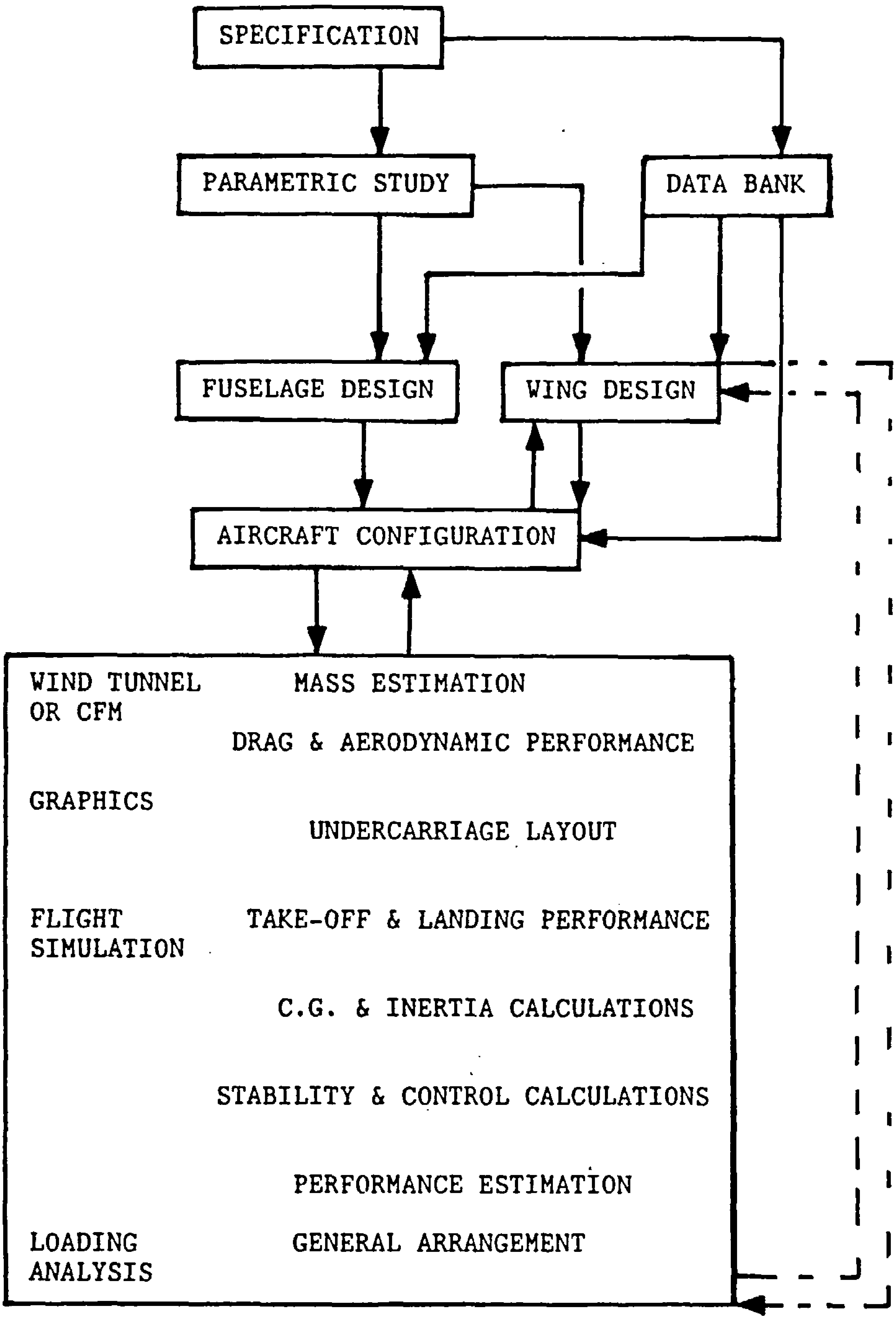


FIGURE 3.1 MAJOR STAGES IN AN AIRLINER INITIAL DESIGN PROCESS

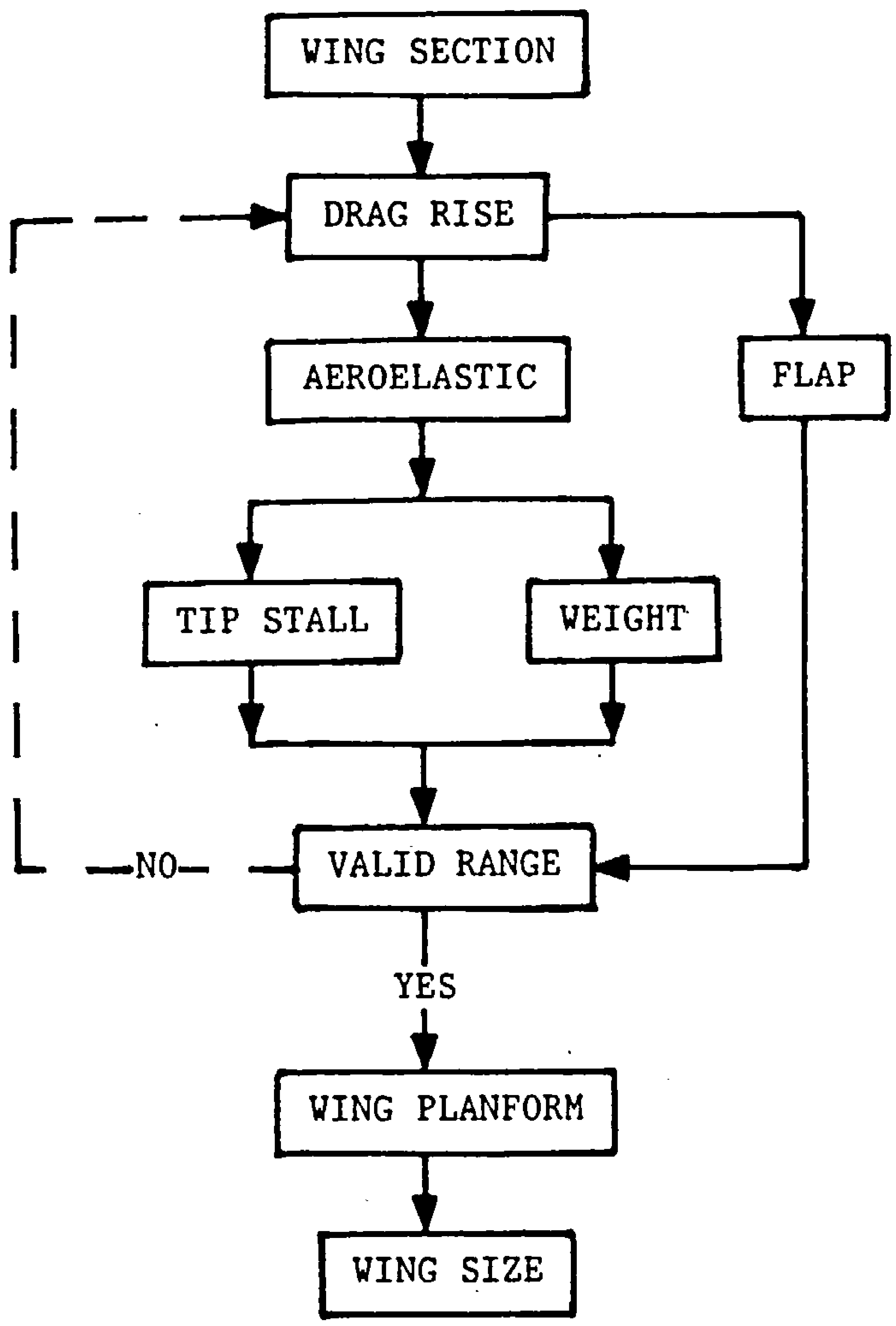


FIGURE 3.2 WING DESIGN STEPS

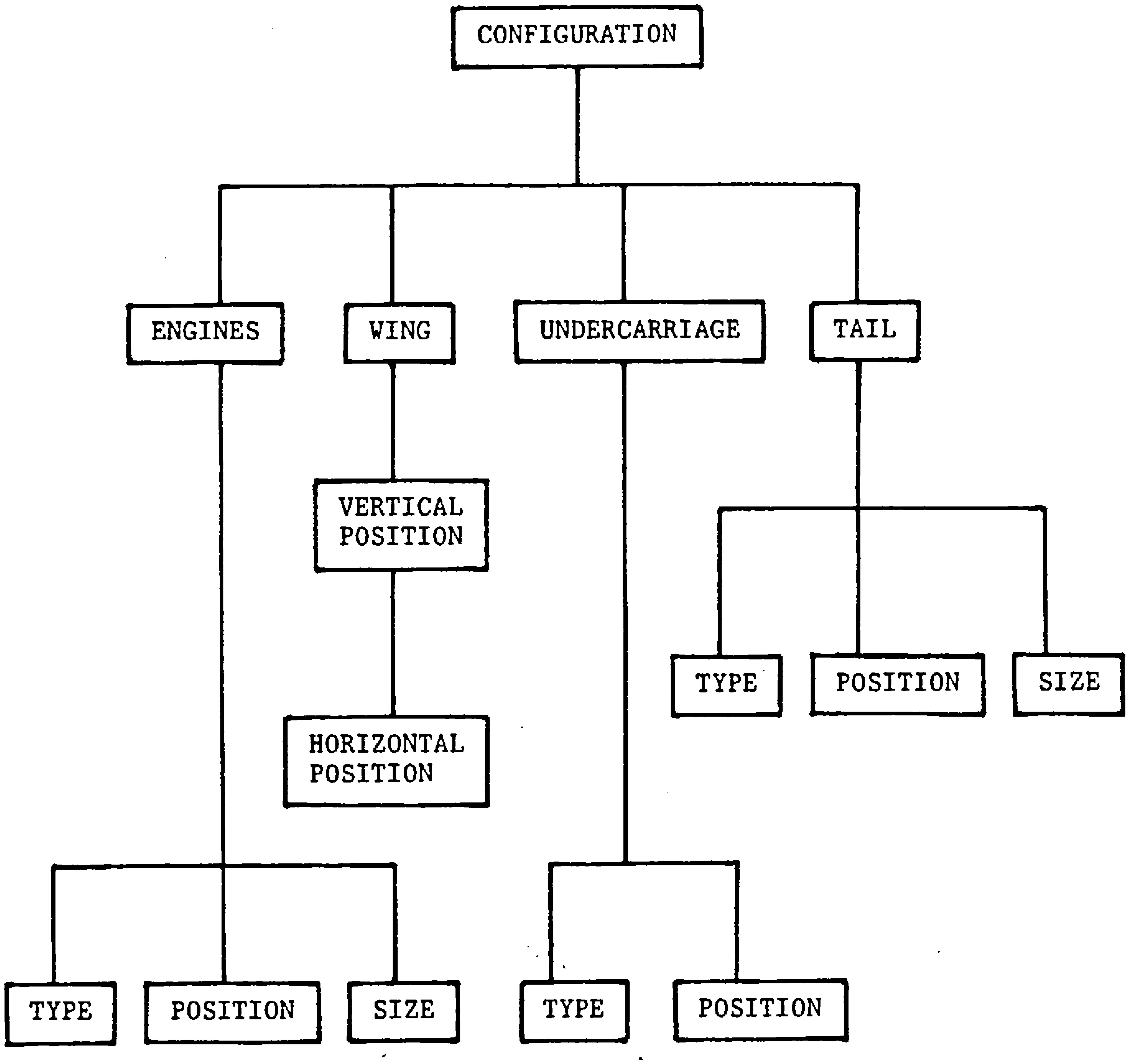


FIGURE 3.3 SIMPLIFIED AIRCRAFT CONFIGURATION

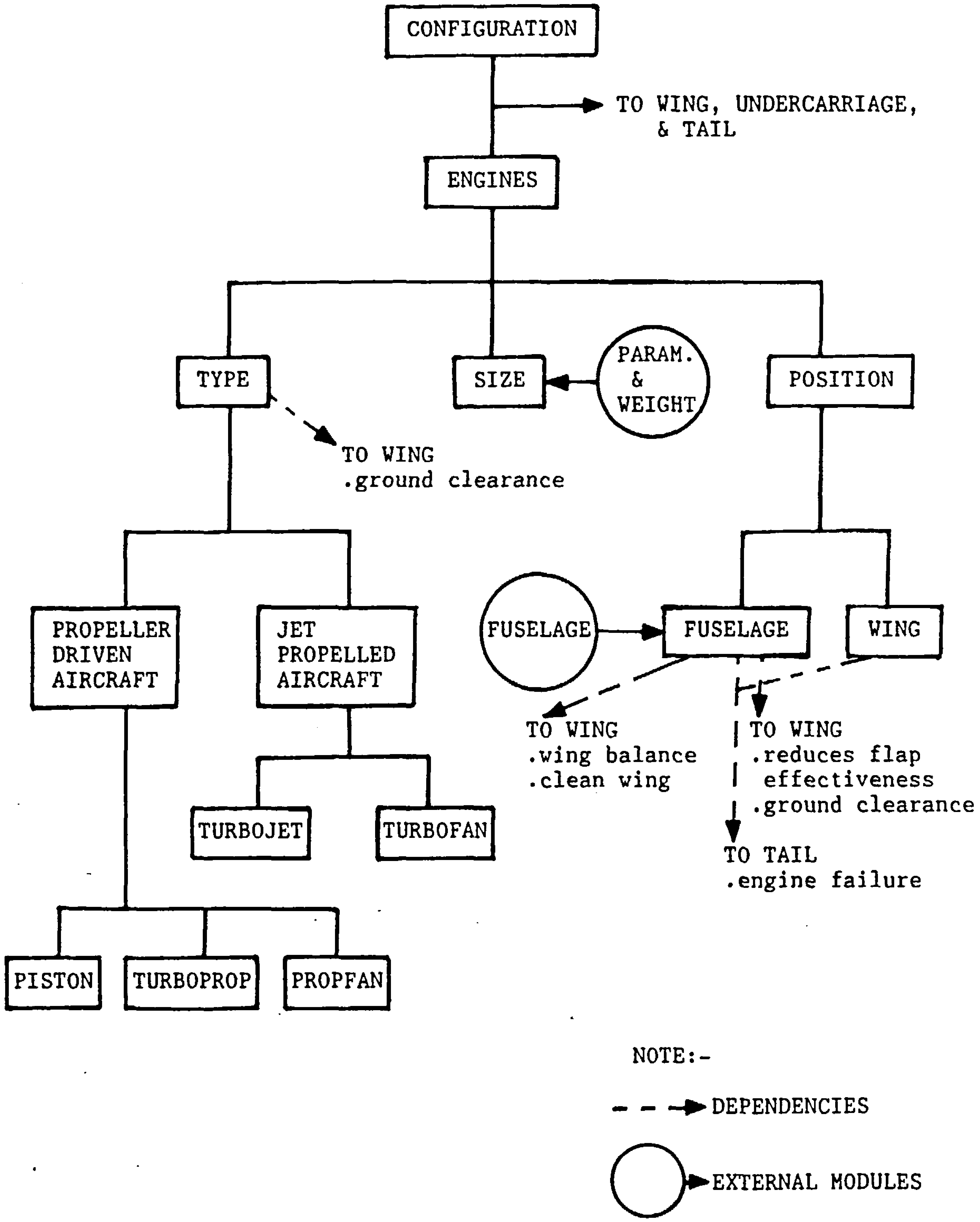


FIGURE 3.4 ENGINE CONFIGURATION

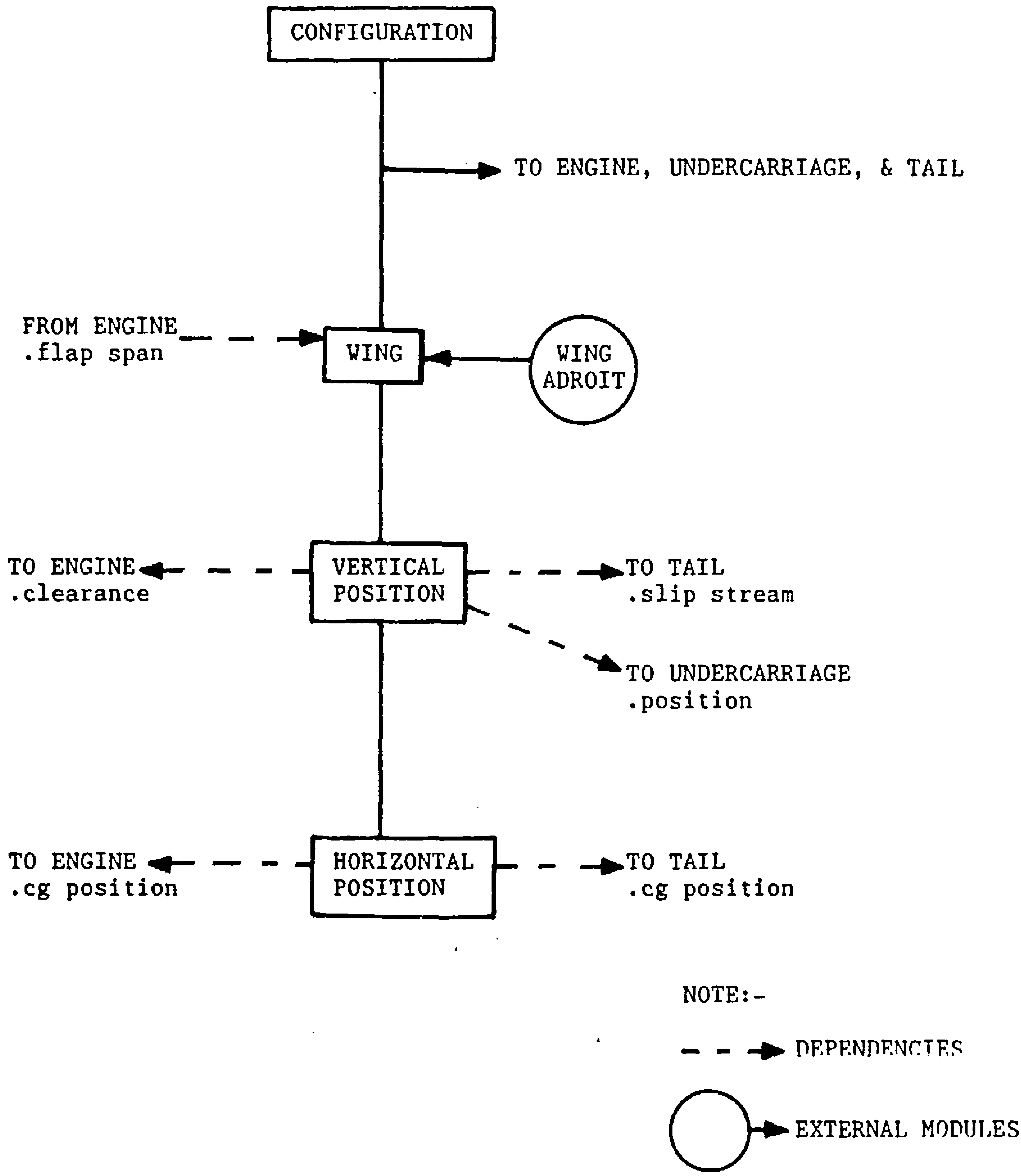


FIGURE 3.5 WING CONFIGURATION

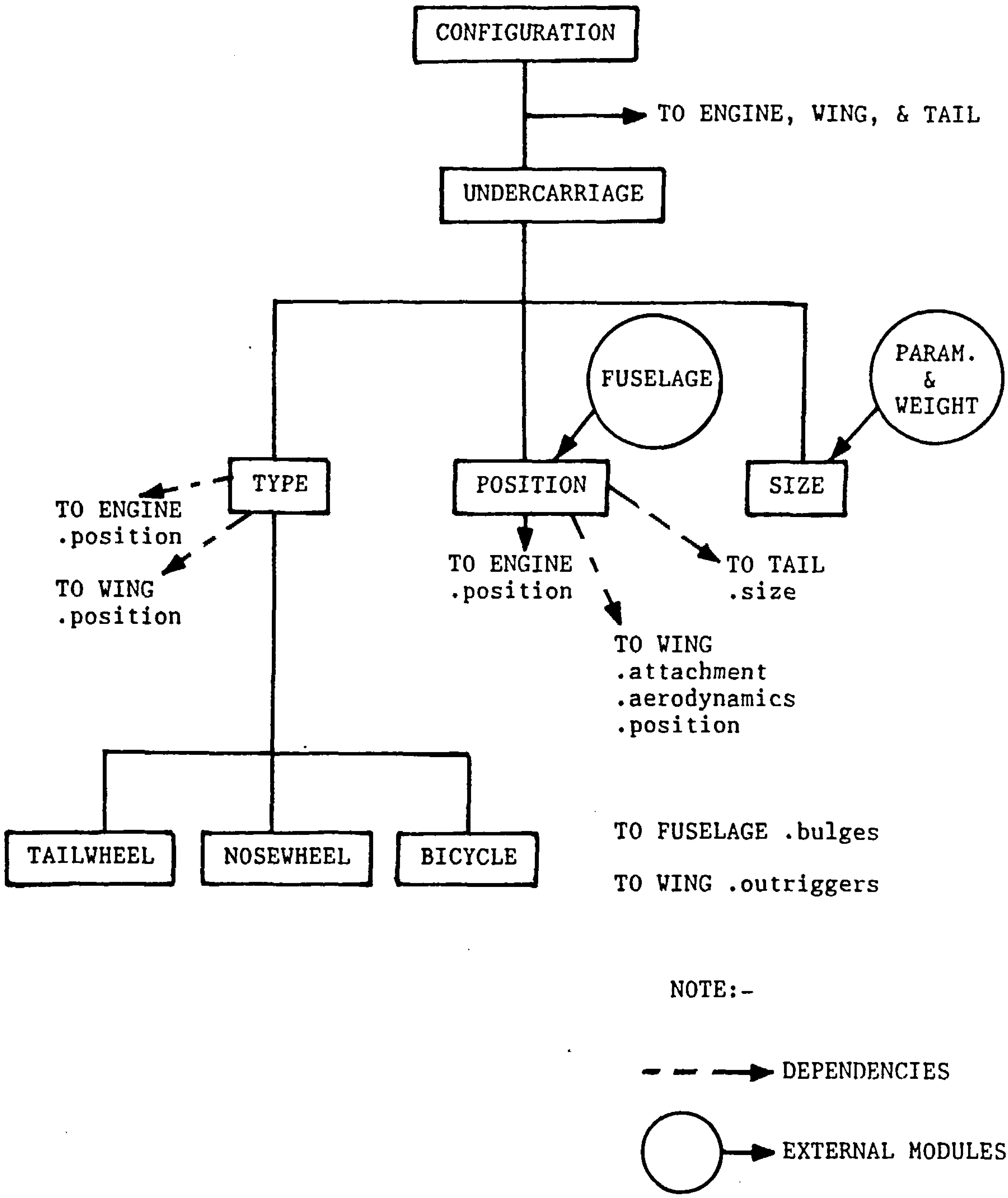


FIGURE 3.6 UNDERCARRIAGE CONFIGURATION

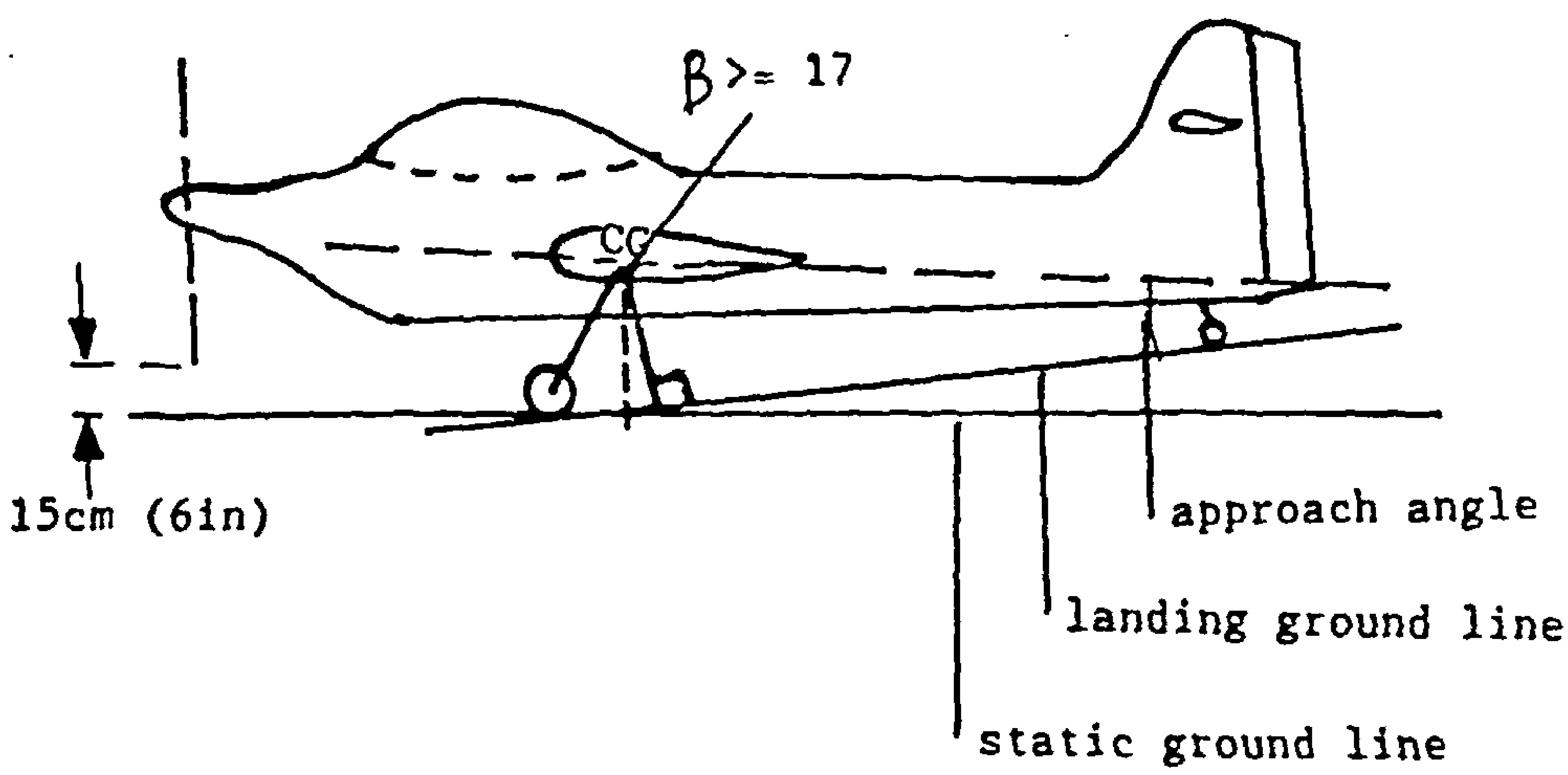


FIGURE 3.7 TAILWHEEL LAYOUT IN ELEVATION

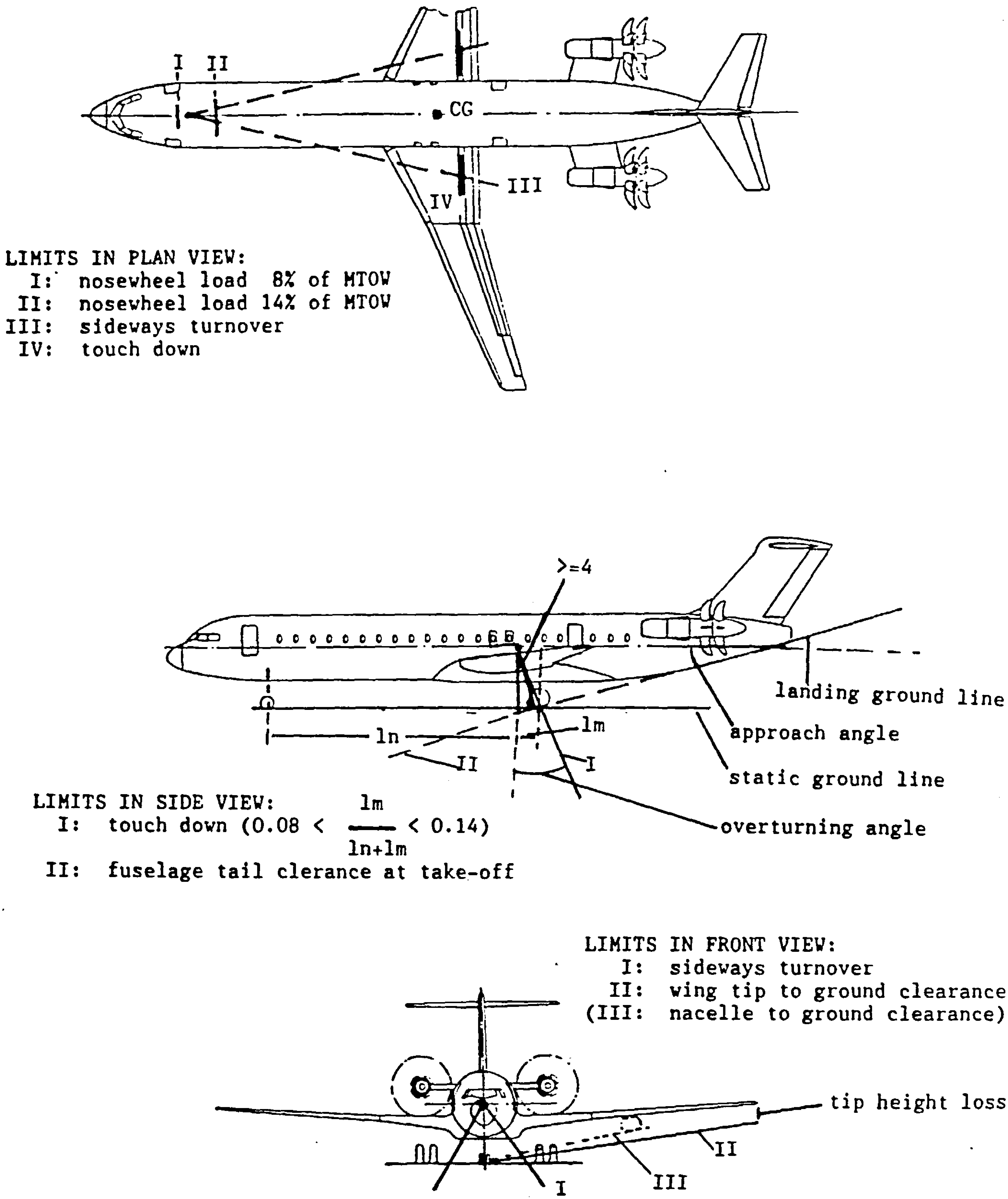


FIGURE 3.8 WHEELS DISPOSITION IN ELEVATION

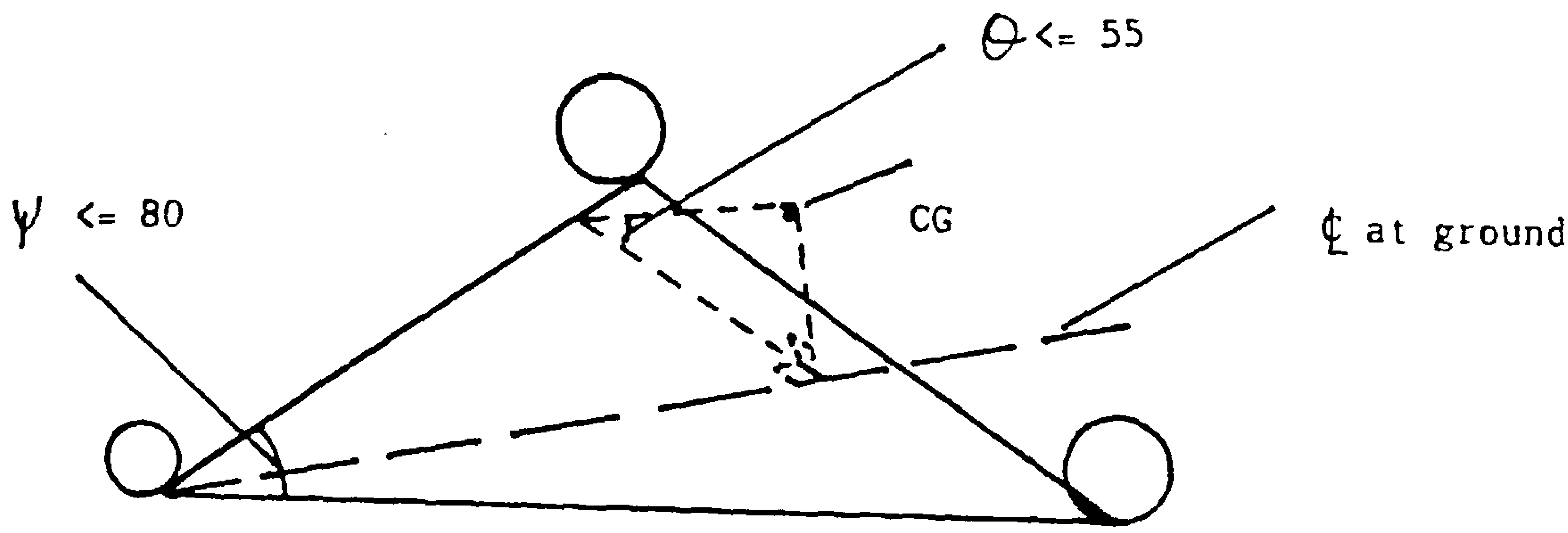


FIGURE 3.9 UNDERCARRIAGE LAYOUT IN PLAN

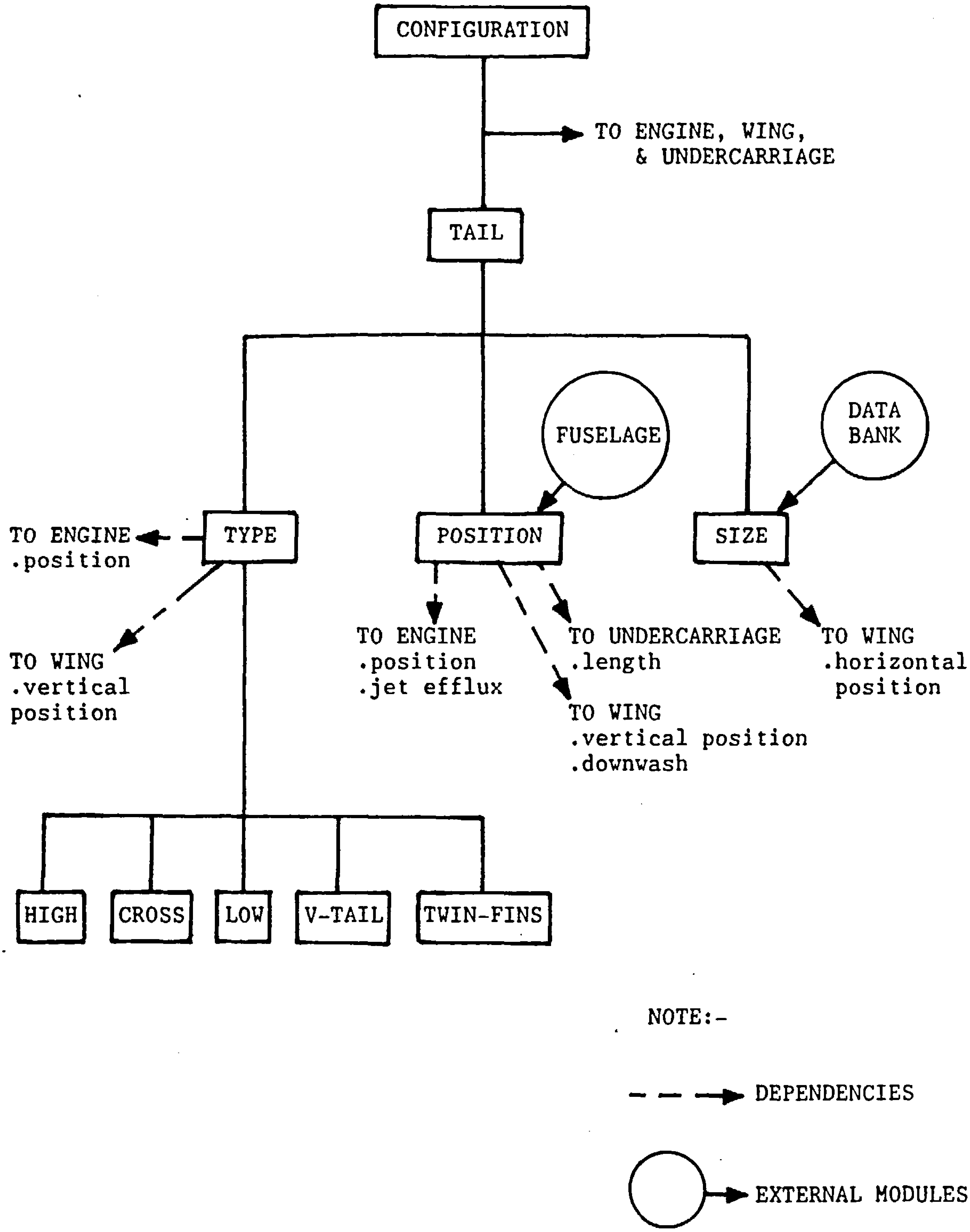


FIGURE 3.10 TAIL CONFIGURATION

LAYOUT AIRCRAFT	WING	ENGINE	UNDERCARRIAGE	TAIL
EXECUTIVE JET	low-swept-wing; good ditch,crash, speed & aesthetics	3 fuselage mounted engines for; clearance, safety, & low noise	tricycle with single main legs, each with one or two wheels	high-tailplane for good clearance
COMMUTER (propeller engines)				
COMMUTER (jet engines)				
AIRLINER (over-land)				
AIRLINER (over-water)				
CARGO AIRCRAFT				

FIGURE 3.11 MATRIX DISPLAY OF AIRCRAFT CONFIGURATION

CHAPTER 4

1ST PASS: WING DESIGN KNOWLEDGE

4.1 INTRODUCTION

In the previous Chapter, the full aircraft design process is described in all its complex nature. It is clear that whilst it would be nice to implement this into an expert system in a single pass, such a move represents too large an advance on current knowledge to be wise. The better approach which is well suited to the present research objective is to develop some initial ideas with a limited implementation on a restricted part of the total design problem.

Wing design is an iterative process, consisting of straightforward computations supplemented with judgemental decisions by the design expert. It represents a small but representative part of aircraft design and as such is an ideal starting point in finding suitable formalisms for structuring complex design knowledge in such a way that the represented problem becomes transparent to the designer and tractable by computer.

The presentation begins by considering the wing design process as described in section 3.6, but the description is directed towards identifying the rules and algorithms used by the expert designer. Based on these aspects, a framework for an aircraft design expert system is defined. A prototype program (27, 28, 29) for wing design called ADROIT (Aircraft Design by Regulation Of Independent Tasks) has been implemented using this framework, and whose scope, procedures and mode of operation are to be described in Chapter 6. However, for the purpose of this Chapter, reference is made to the status of the framework within this implementation together with illustrations of the knowledge representation features used.

4.2 ASPECTS RELEVANT TO KNOWLEDGE REPRESENTATION

Section 3.6 described wing design in terms of its functionality. The purpose of this section is to identify the knowledge representation aspects of the wing design process in terms of the algorithms and heuristics used by the designer.

4.2.1 1st STAGE: two-dimensional wing design

The expert designer selects a wing section from a set of similar sections according to the aircraft specification. For a high subsonic airliner this comparison is based on the theoretical and/or experimental data corresponding to the stall characteristics at low speeds, and the lift/drag ratio, pitching moment, and 2-D drag rise at cruise conditions. This comparison forms part of the qualitative aspects of the design which cannot be given in algorithmic form but rather in terms of concepts which are based on as yet unformalised experience and judgement.

The specification sometimes lies outside the experimental data available and suitable approximations must be made. In the theoretical case, the calculation procedure used is chosen from a pool of available procedures for specific classes of wing sections. In both cases, the expert uses his experience and intuition to select the appropriate method.

4.2.2 2nd STAGE: three-dimensional effects

For a high subsonic airliner the 3-D drag rise, aeroelastic stiffness, tip stall, flap effectiveness, and wing weight need considering at this stage to achieve a suitable design.

4.2.2.1 3-D drag rise - is evaluated using appropriate procedures based on expert judgements about engine position, sweep angle, t/c ratio, and wing-fuselage interation.

4.2.2.2 Aeroelastic stiffnesss - The designer specifies the simulation model to be used in order to evaluate the mechanical properties of the wing. In a first iteration, an estimation of these properties using simple procedures suffices whereas in later iterations, the greater precision of the Finite Element (FE) method (74) is required.

4.2.2.3 Tip stall - The wing spanwise load distribution is evaluated using appropriate procedures, the results of which are presented in a graphical form for interpretation upon which the designer may decide to improve the loading characteristics (to avoid stall) by a combination of twist, taper, and camber.

4.2.2.4 Flap effectiveness - The designer selects the flap and other high lift devices in order to obtain the required landing lift coefficient.

4.2.2.5 Wing weight estimate - uses a variety of algorithms which take into account the construction materials and technology employed. The designer uses his judgement to select an appropriate method based on the information available and interpret the results found.

4.2.3 3rd STAGE: wing planform

The general shape of the candidate wings as viewed from the top are compared against available wing designs. The designer uses his judgement to select the most promising candidates and modify design parameters if necessary.

4.2.4 4th STAGE: design for take-off and landing

The designer uses suitable procedures to check that the take-off and landing performance set in the parametric study are satisfied using the selected high lift devices.

4.2.5 5th STAGE: final sizing

The final sizing procedures use an estimate of the total aircraft weight evaluated from other parts of the overall design process to scale the wing, obtaining an appropriate lift parameter for the aircraft.

4.2.6 6th STAGE: iteration

The above design steps are repeated with information from previous design paths. In some cases an iteration will not cover the design cycle but will jump back to a previous design step. The repetition of a design step in a later iteration may involve the engagement of subprocesses different from those in previous iterations.

4.3 A FRAMEWORK FOR AN AIRCRAFT DESIGN EXPERT SYSTEM

The wing design process has been described in terms of the general procedures and judgements used by the designer. The present section outlines a framework for building an aircraft design expert system (18) based on the wing design requirements. The framework consists of six modules as shown in Figure 4.1: user interface, templates, global controller, method bank, component information database, and design representation. The functional specification of each module is presented together with its status and the knowledge representation aspects within the implementation.

4.3.1 User interface

DEFINITION

The user interface is responsible for the interaction between the user and the system through the global controller. It must be capable of both requesting and providing information, including expert advice and explanations of the reasoning structure employed in the design. There is also a requirement to be able to present graphical information on results commensurate with the designer's needs. This capability must be able to track the user and provide appropriate explanation and tutoring facilities dependent upon the systems perception of the users level of competance.

IMPLEMENTATION

A simple user interface has been developed to control the interaction between the user and the system. When the consultation is started the user is asked questions until the evaluation of all the templates is completed. The results and conclusions are presented as dictated by the requirements of each template. Some of the facilities available to the user include:

1. The data input is performed interactively as required by a design step.

2. Allows the user to specify the order in which the different design steps are evaluated.

3. Checks for wrong input and issues the appropriate error message.

4. At any time during a consultation the user has a set of commands to control the extend of a consultation.

5. The interpretation and presentation of results are an important requirement from any design process. ADROIT provides clear and self explanatory output through the use of text, tables, or graphics.

4.3.2 Templates

DEFINITION

A central task within the proposed framework is the representation of the design knowledge. Experience in solving problems has taught designers that the best way to solve a problem is to break it up into smaller and independent problems also, experience influences their choice of methodology in dealing with problems. The template concept has been introduced to encapsulate the design knowledge. Two types of templates are proposed:

Action templates represent independent design steps and are constructed according to the following scheme.

1. An identifier which allows the template to be identified uniquely.

2. A type which describes the broad functions of the template to the user.

3. An input section describes user supplied and reference data required by the template.

4. A control script sequences the algorithms and rules used by the template. The algorithms denote procedures and inferences used in the design that guarantee a solution to a problem while the rules represent the strategies used by expert designer.

5. The byproduct displays intermediate results to the user.

6. The output section displays information to the user and which is to be passed to other templates.

Control templates are used to abstract the design process by having a number of related design steps represented as a single design step or stated differently, the design tree can be simplified by collapsing one or more nodes into a single node.

IMPLEMENTATION

The relation between the different wing design steps can be represented as a partially ordered tree of templates. Figure 4.2 shows the action and control templates corresponding to the wing design implementation. Wing design is represented by two control templates namely, wing_section and sweep_angle. Wing_section consists of a single action template of the same name, whereas sweep_angle consists of five action templates; drag_rise_3d, aeroelastic, tip_stall, flap, and wing_weight. Within this tree, subgoal-supergoal, conjuctive, and sequential relationships between design steps are captured. The following PROLOG facts represent the step sequencing i.e., must_follow(STEP1,STEP2) indicates that STEP1 follows STEP2.

```
must_follow(sweep_angle,  wing_section).
must_follow(flap,         drag_rise_3d).
must_follow(aeroelastic,  drag_rise_3d).
must_follow(tip_stall,    aeroelastic).
must_follow(wing_weight,  aeroelastic).
```

The following PROLOG facts represent the type of each template i.e., type(STEP,TYPE) indicates STEP is of TYPE action or control.

```
type(wing_design,      control).
type(wing_section,     action).
type(sweep_angle,      control).
type(drag_rise_3d,     action).
type(flap,             action).
type(aeroelastic,      action).
type(wing_weight,      action).
type(tip_stall,        action).
```

A description of the aeroelastic action template follows in terms of PROLOG facts and rules. This simple example shows how PROLOG can deliver programs which are both concise and clear, because of the generality of the rules used and the declarative nature of the language. The definition of an action template follows:

```
Display introduction text
Request input from the user
Execute the algorithms and rules for the template
Display byproducts
Display outputs
```

The PROLOG equivalent statements are:

```
template(_template) :-
  intro(_template),
  input(_template),
  control(_template),
  byproduct(_template),
  output(_template).
```

Where the PROLOG variable _template is bound to the name aeroelastic and the above statements are executed in the order given. Each of these statements constitutes a PROLOG goal which must be satisfied within the database.

4.3.2.1 Introduction - The PROLOG goal intro(aeroelastic) introduces the aeroelastic template in the following way:

```
Choose the template heading
Choose the template introduction text
Display heading
Display text
```

The PROLOG equivalent statements are:

```
intro(_template) :-
  intro_heading(_template, _heading),
  intro_text(_template, _text),
  print_heading(_heading),
  print_text(_text).
```

The intro_heading and intro_text goals are particular to each template and are defined as PROLOG facts.

4.3.2.2 <u>Input</u> - The goal input(aeroelastic) requests from the user the engine position, cruise altitude, aspect ratio, and whether or not active controls are to be used:

```
Choose the input parameters for template
While the list of input parameters is not empty do
  Choose a parameter from the list of parameters
  Request input corresponding to the parameter
  Store answer in the database
  Remove parameter from the list of parameters
```

The PROLOG equivalent statements are:

```
input(_template) :-
  in(_template, _parameters),
  get_input(_parameters).
```

Where the goal in(aeroelastic, _parameters) returns the list of input parameters for the aeroelastic template using:

```
in(_template, _parameters) :-
  bagof(_parameter,
        input_of(_parameter, _template),
        _parameters).
```

For each template the input parameters are defined as PROLOG facts and for the aeroelastic template these are:

```
input_of(altitude,        aeroelastic).
input_of(aspect_ratio,    aeroelastic).
input_of(engine_position, aeroelastic).
input_of(active_controls, aeroelastic).
```

The definition of the goal get__input is given by:

```
get_input([]).
get_input([_parameter | _parameters]) :-
  ask(_parameter, _value),
  assert(user_input(_parameter, _value)),
  get_input(_parameters).
```

Where the goal ask displays the question corresponding to _parameter, and and returns the answer _value to be stored in the database in the fact user_input.

4.3.2.3 <u>Control</u> - The goal control(aeroelastic) executes three algorithms and three rules:

```
Choose the output parameters for this template
While the list of output parameters is not empty do
  Choose a parameter from the list
  Execute the corresponding rule or algorithm
  Store the results in the database
  Remove the parameter from the list of parameters
```

The PROLOG equivalent statements are:

```
control(_template) :-
  out(_template, _parameters),
  actions(_template, _parameters).
```

Where the goal out(aeroelastic, _parameters) returns the list of output parameters for the aeroelastic template using:

```
out(_template, _parameters) :-
  bagof(_parameter,
        output_of(_parameter, _template),
        _parameters).
```

For each template the output parameters are defined as PROLOG facts and provide the name of the items to be evaluated. For the aeroelastic template these are:

```
output_of(bending_relief, aeroelastic).
output_of(load_factor,    aeroelastic).
output_of(vd,             aeroelastic).
output_of(torsion,        aeroelastic).
output_of(bending,        aeroelastic).
output_of(valid_angles,   aeroelastic).
```

The definition of the goal actions is:

```
actions(_, []).
actions(_template, [_parameter | _parameters]) :-
  action(_template, _parameter, _data),
  assert(data(_template, _parameter, _data)),
  actions(_template, _parameters).
```

The goal action(aeroelastic, _parameter, _data) executes the rule (r_parameter) or algorithm (a_parameter) associated with _parameter and returns the values found in the variable _data. For the aeroelastic template these are:

```
action(aeroelastic, bending_relief, _data) :-
  r_bending_relief(_data).
action(aeroelastic, load_factor,    _data) :-
  r_load_factor(_data).
action(aeroelastic, vd,             _data) :-
  a_vd(_data).
action(aeroelastic, torsion,        _data) :-
  a_torsion(_data).
action(aeroelastic, bending,        _data) :-
  a_bending(_data).
action(aeroelastic, valid_angles,   _data) :-
  r_valid_angles(_data).
```

Although there are no programming differences between a rule and an algorithm, conceptually a rule captures some piece of expertise while an algorithm uses a general and well known procedure to evaluate a value. For example, the algorithm used to evaluate the diving speed is:

```
Get user defined cruise altitude in feet
Get user defined maximum cruise Mach number
Compute temperature at cruise altitude
Compute speed of sound at temperature
Compute diving speed in Knots
```

The PROLOG equivalent statements are:

```
a_vd(_VD) :-
  user_input(cruise_altitude,          _H),
  user_input(maximum_cruise_mach_number, _Mmax),
  temperature(_H, _T),
  speed_of_sound(_T, _a),
  _VD = (_Mmax + 0.05) * _a.
```

```
temperature(_H, _T) :-
  _H <= 36000,
  _T = 288.2 - 0.00198 * _H.
temperature(_H, 216.7) :-
  _H > 36000.

speed_of_sound(_T, _a) :-
  _a = (1.9438 * sqrt(401.8 * _T)).
```

An example of a rule is that used by the expert to obtain the valid sweep angles based on the torsion and bending stiffness checks. The following statements summarised the actions performed:

```
Choose list of sweep angles As
Choose target aspect ratio AR (user input)
Choose list of torsion aspect ratios AR1s
Choose list of bending aspect ratios AR2s
Compute tolerance T (within 10% of AR)
While list of angles is not empty
  Choose an angle A from As
  Choose the corresponding aspect ratio AR1 from AR1s
  Choose the corresponding aspect ratio AR2 from AR2s
  Compare AR1 and AR2 against T
  If T is satisfied add A to list of valid angles
  Remove A from As
```

The PROLOG equivalent statements are:

```
r_valid_angles(_valids) :-
  angles(_As),
  user_input(aspect_ratio, _AR),
  data(aeroelastic, torsion, _AR1s),
  data(aeroelastic, bending, _AR2s),
  tolerance(aspect_ratio, _AR, _T),
  bagof(_A,
        (member(_A, _As),
         locate(_A, _As, _N),
         locate(_AR1, _AR1s, _N),
         locate(_AR2, _AR2s, _N),
         _AR1 >= _T,
         _AR2 >= _T),
        _valids).
```

4.3.2.4 Byproduct - The goal byproduct(aeroelastic) displays the torsion and bending stiffness results for the range of sweep angles.

```
Choose byproduct heading
Choose byproduct text
```

```
Display heading
Display text
Display torsion stiffness aspect ratios and sweep angles
Display bending stiffness aspect ratios and sweep angles
```

The PROLOG equivalent statements are:

```
byproduct(_template) :-
  byproduct_heading(_template, _heading),
  byproduct_text(_template, _text),
  print_heading(_template, _heading),
  print_text(_template, _text),
  ans_byproduct(_template).
```

4.3.2.5 Output - The goal output(aeroelatic) displays the valid sweep angles for both the torsion and bending stiffness checks.

```
Choose output heading
Choose output text
Display heading
Display text
Display valid angles
```

The PROLOG equivalent statements are:

```
output(_template) :-
  output_heading(_template, _heading),
  output_text(_template, _text),
  print_heading(_template, _heading),
  print_text(_template, _text),
  ans_output(_template).
```

4.3.3 Global controller

DEFINITION

The global controller maintains control of the potential actions awaiting execution, determines which pending action should be executed next, executes the chosen action, and attempts to maintain a consistent representation of the emerging solution.

IMPLEMENTATION

ADROIT will lead the user through the details of each template as they are reached. When the user has completed a template the validity of the results is checked. If unacceptable results have been generated by a template or the results are incompatible with other templates then backtracking to one or more precursor templates of the failed template must be carried out in order to correct the design. This is performed by allowing the user to re-evaluate the failed template and its precursors. For example, if the user has stepped outside the limits of the current design knowledge by asking ADROIT to over-extrapolate one of its graphs of wing section aerodynamic data then, ADROIT will warn the user and ask which of the previous templates should be modified. For each template to be re-evaluated the user can change the input values. Once new values have been selected ADROIT will offer the user the opportunity to repeat that step or to go on to a different point in the design. The design steps left incomplete will be offered again, when the appropriate part of the design tree is reached. If ADROIT has no built-in validity checks for a template, it asks the user whether or not the results are valid.

ADROIT represents the relationship between the different templates as the PROLOG facts described above, and uses the following relationship to determine the ancestor of each step.

```
always_after(_s1, _s2) :- must_follow(_s1, _s2).

always_after(_s1, _s2) :-
        must_follow(_s3, _s2),
        always_after(_s1, _s3).
```

The first PROLOG rule is used to see if step _s1 is always after _s2 from the fact must_follow(_s1, _s2). The second rule, checks that _s1 is after _s2 if there is a step _s3 which follows _s2 and _s1 is after _s3.

In determining a problem with a design step ADROIT stores the possible checks as PROLOG facts and checks that these are not true using PROLOG rules. For example, to determine if over-extrapolation has occurred during the wing section choice the following fact and rules are used.

```
only_checks_for(wing_section,  extrapolation).

problem_with(extrapolation,'Over-extrapolation: Lift/drag ratio') :-
    data(cd, _, _cd),
    not(between(_cd, 0.0092, 0.02481)).

problem_with(extrapolation,'Over-extrapolation: Pitching moment') :-
    data(cm, _, _cm),
    not(between(_cm,-0.21,-0.047)).

problem_with(extrapolation,'Over-extrapolation: 2D drag rise') :-
    data(md, _, _md),
    not(between(_md,0.63,0.88)).
```

The fact only_checks_for(STEP,PROBLEM) stores the possible problems with a step. The rules problem_with(PROBLEM,MESSAGE) with PROBLEM = extrapolation check for over-extrapolation, if it is satisfied MESSAGE is displayed to the user and backtracking is invoked. The fact data(ITEM,SECTION,VALUE) stores the results in the database, and the relation between(VALUE,VALUE1,VALUE1) checks that VALUE is between VALUE1 and VALUE2.

4.3.4 Method bank

DEFINITION

General algorithmic and heuristic procedures employed in the design of aircraft should be readily available from a method bank. An example of a general algorithmic procedure would be the use of a Finite Element module to perform accurate structural analysis.

IMPLEMENTATION

An example of an algorithmic procedure is the sequence of calculations used to assess the aeroelastic stiffness of the wing. A heuristic procedure does not, essentially, employ calculation and an example in ADROIT is the method used to assess the stall handling characteristics of the wing.

4.3.5 Component information database

DEFINITION

The system should be able to interact with a database containing a collection of standard aircraft components or even complete aircraft, airworthiness requirements (e.g., structural envelope) etc., in order to ease the user supplied knowledge.

IMPLEMENTATION

Descriptions of three supercritical aerofoil sections are available as shown below but there is no general component database as yet.

The wing section aerodynamic data is stored as PROLOG facts. For example, in the case of the drag coefficient facts with four arguments are used as shown below.

```
datav(cd,'RAE 9515',0.4,[1.088,-2.047,1.262,-0.243]).
datav(cd,'RAE 9515',0.45,[0.016,-0.018,0.017]).
datav(cd,'RAE 9515',0.5,[0.679,-1.278,0.788,-0.146]).
datav(cd,'RAE 9515',0.55,[-0.515,0.983,-0.613,0.139]).
datav(cd,'RAE 9515',0.6,[0.05994,-0.068687,0.0342341]).
```

The first argument is the name of the data cd (drag coefficient), the second argument is the name of the section (RAE 9515), the third argument is the name of the curve (lift coefficient), and the fourth argument represents a list containing the polynomial coefficients of the curve x (cruise Mach number) against y (the unknown = drag coefficient). ADROIT will extrapolate or interpolate between these curves (lift coefficients) as necessary.

4.3.6 Design representation

DEFINITION

The system should store the current design represented as a collection of components. Here, the system must be capable of reasoning about incomplete specifications, deal with design constraints, and represent the design history.

IMPLEMENTATION

This capability has not been implemented in the ADROIT program. Although, the user replies, the results evaluated by the templates, and the status of each design step (i.e., done or not done) are stored in the database.

4.4 CONCLUSIONS

A framework, comprising of six separate modules, for an aircraft design expert system has been presented based on the initial knowledge representation requirements of aircraft design. The formulation has been based on the wing design problem, a smaller and representative part of the aircraft design process, which has been implemented using this framework. The status of each module within the implementation together with particular knowledge representation features have been presented.

The following Chapter outlines the knowledge representation requirements for aircraft configuration and lists the extensions necessary to the present architecture in order to deal with this module. While Chapter 6 provides a detail account of the implementation in terms of the procedures used, mode of operation, and a record of a consultation session.

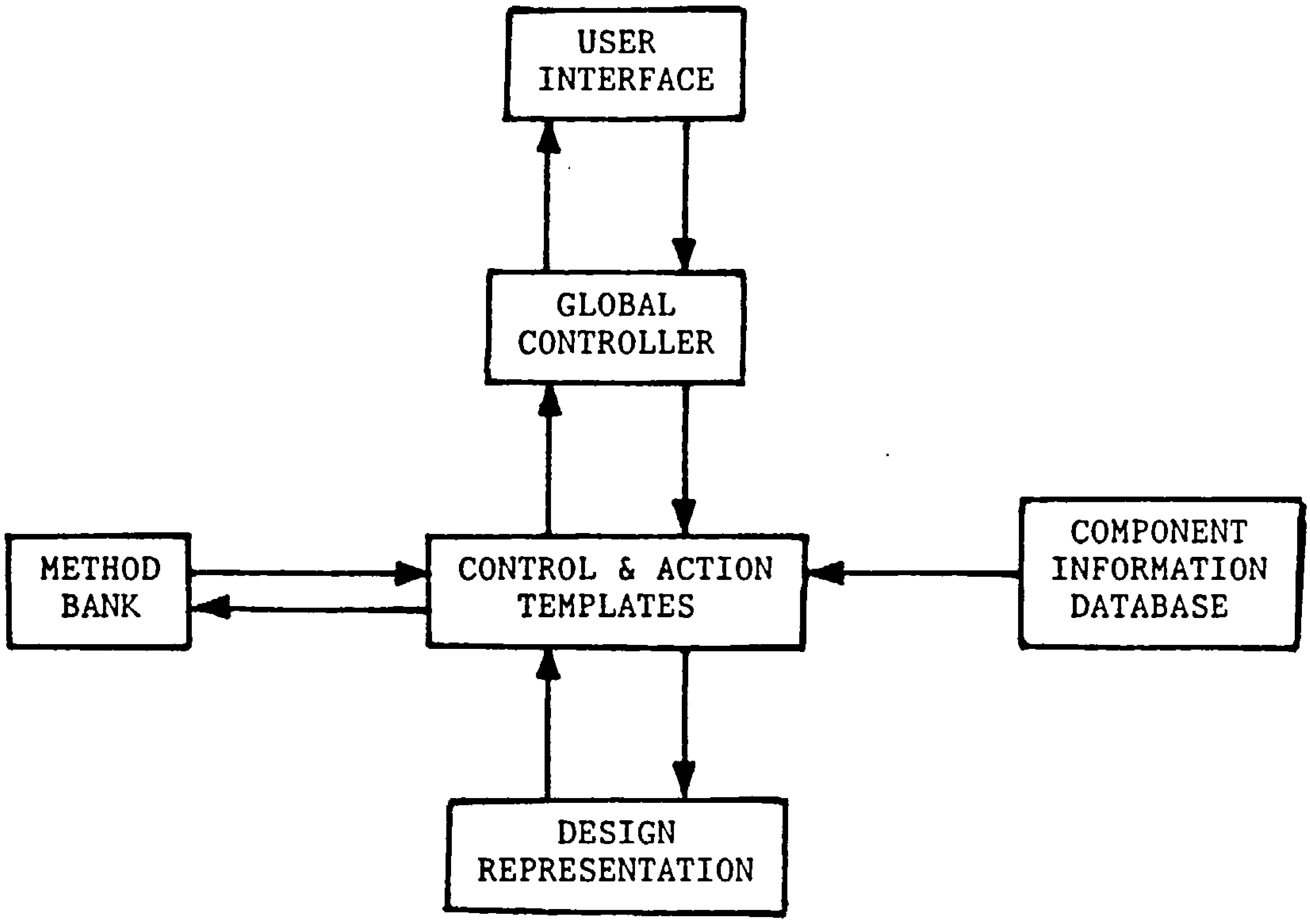


FIGURE 4.1 AIRCRAFT DESIGN EXPERT SYSTEM FRAMEWORK

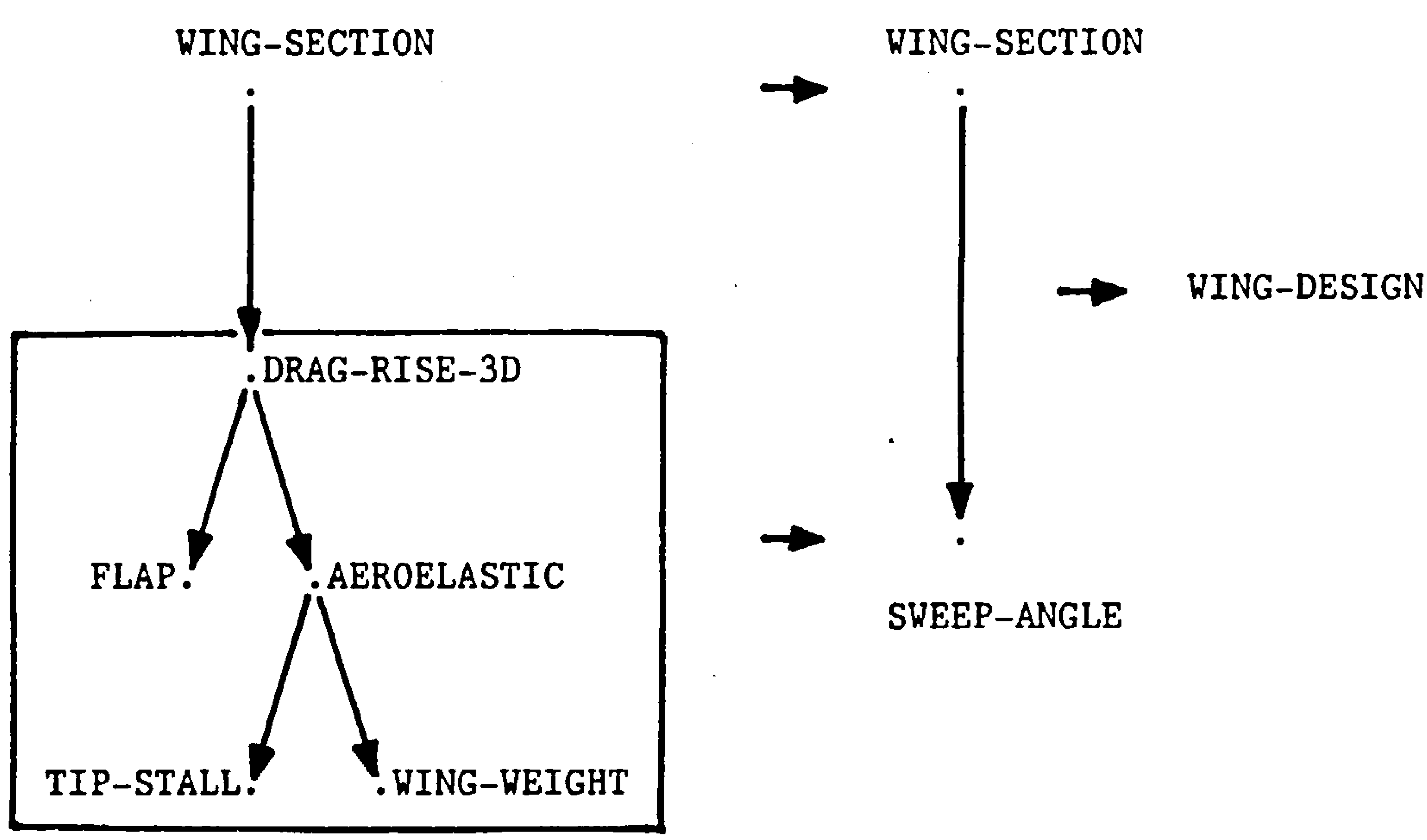


FIGURE 4.2 WING DESIGN TEMPLATE REPRESENTATION

CHAPTER 5

2ND PASS: AIRCRAFT CONFIGURATION KNOWLEDGE

5.1 INTRODUCTION

Chapter 4 successfully represented the knowledge associated with the limited problem of wing design. In order to increase the level of complexity, the present Chapter considers the aircraft configuration problem within which wing design is but one aspect. The aircraft configuration problem under consideration, described in Chapter 3, constitues a major step within aircraft design on which the success of the aircraft to be built is of decisive importance. It refers to the general layout, external shape, dimensions etc., as characterised by the relative location of the main components (i.e., wings, engines, tail surfaces, and undercarriage) based on the investigation into and interpretation of the aircraft function and a translation of the most pertinent requirements. Three essential characteristics of aircraft configuration are; first, it is always made up of iterations thus, after a trial configuration has been subjected to a first analysis of its characteristics (weight, performance, flying qualities etc.) it will be seen either that it does not meet all the requirements, or that it does comply with them but improvements in some respects are possible; second, interactions between the aircraft components occur in such a way that when a choice is taken for a particular component according to the specification an understanding of the other components behaviour is necessary; finally, no clear cut procedure can be followed, much of which relies on the experience and resources of the designer in order to compare and judge different configurations according to an aircraft specification.

This Chapter describes the main details of the resulting output during the second pass towards an aircraft design expert system. The types of knowledge found in aircraft design have been identified in Chapter 2 and are used in the present Chapter to aid in identifying the knowledge representation requirements for aircraft configuration. Various aircraft configuration examples for each type of knowledge are provided. Based on this study, the conceptual differences between aircraft configuration and wing design are presented together with the necessary extensions to the aircraft design expert system framework proposed in Chapter 4.

5.2 ASPECTS RELEVANT TO KNOWLEDGE REPRESENTATION

The different aspects which have to be taken into account when developing and evaluating different knowledge representation formalisms have been indicated in Chapter 1. Aircraft configuration as shown in Chapter 3 represents a design step in which a large and varied amount of knowledge is used by the designer to arrive at a suitable solution by interpreting the different and often conflicting requirements.

In order to define, implement, and refine the aircraft configuration problem examples of the types of knowledge found are given so as to improve the conceptualisation and transparency of the problem by separating the domain specific knowledge from the general problem-solving knowledge.

5.2.1 Commonsense

Within aircraft configuration this type of knowledge is required in order to draw conclusions from partial information and make assumptions as a resource limiting process by default or subject to certain conditions. For example,

```
[IF     large transport aircraft and
        low wing
 THEN   use tricycle undercarriage]
**CONCLUSION: light undercarriage structure and easy to retract

[ASSUME  double slotted flaps]

[ASSUME  high wing
 IF      large cargo aircraft]

[ASSUME  underwing engines
 UNLESS  ground clearance problems]
```

Typically, a designer draws new information by exploring tentative solutions. For example,

[1) Suppose we use rear mounted engines
2) then there is ground clearance
3) and there is a clean wing

4) but there is a large cg movement

5) So, we would better use wing mounted engines]

5.2.2 Constraints

Throughout aircraft configuration constraints are frequent and provide a powerful way to reduce the number of choices available at a design point and help to maintain the consistency of the solution. For example, in the selection of an engine type the designer has to take into account various factors, within these the use of the cruise Mach number (M) provides an initial selection method.

[IF	$0.2 < M < 0.5$	THEN	piston]
[IF	$0.3 < M < 0.65$	THEN	turboprop]
[IF	$0.7 < M < 0.8$	THEN	propfan]
[IF	$0.65 < M < 0.9$	THEN	turbojet]
[IF	$0.7 < M < 0.9$	THEN	turbofan]

5.2.3 Subproblem interaction

Aircraft configuration constitutes the positioning of the aircraft main components, each of which can be further subdivided into smaller design steps as shown in Figure 3.3. A designer when considering any one of these components must constantly refer to the other components in order to preserve the overall consistency of the solution. For example,

[IF fuselage mounted engines

THEN use high tailplane and
increase tailplane size and
move wing aft]

The above example shows that fuselage mounted engines require the wing to be moved aft and a high tailplane with increased size in order to provide sufficient tail moment arm. In some cases there is a 'spiral' or 'snow ball' effect for example, a cg position which is too far aft can be corrected by moving the wing in the same direction. That is,

[If the wing is moved aft the cg moves in the same direction. The weight moved (wing, fuel, landing gear, and engines) represent say 50 per cent of the total weight and if the cg position is 50 cms too far aft it would appear that by moving the wing 1 metre aft would correct the condition. However, by doing this the tail moment arm is reduced by 1 metre thus the size of the tail surfaces must be increased in order to provide the same effectiveness. This not only adds to weight and drag of the aircraft but because it adds weight at a distance far aft of the cg it again moves the cg aft].

On the other hand, to correct a cg position that is too far forward the following spiral effect occurs,

[If the wing is moved forward, this shift in wing position adds to the efficiency of the aircraft by requiring a smaller tail. This smaller tail will decrease the weight and drag of the aircraft which will again move the cg forward].

5.2.4 Metaknowledge

Aircraft configuration is a highly qualitative design step where decisions are interdependent and the problem-solving strategy employed by the designer is difficult to formalise. Metaknowledge is of particular importance as it embodies the design strategy which depends on the specification and the designer's experience and judgement. Different uses of this knowledge are listed below.

1. Typical of aircraft configuration is the choice which have to be made with conflicting goals. When this situation is detected a number of actions are available; first, generate a new type of solution which satisfies all goals; second, relax the least important goal; finally, generate intermediate solutions where all important goals are satisfied. For example,

    [DO engine type
    BEFORE engine position]

    [DO engine position
    BEFORE selecting a tail position and type]

2. If there are subgoals that are part of several major goals, plan to satisfy those goals before other subgoals. For example, in the undercarriage configuration the following is true

    [DO turnover angle evaluation
    BEFORE pitch and roll limits of aircraft]

3. Check preconditions before executing an operation. For example, within the undercarriage configuration:

   [CHECK elevation requirements
   BEFORE plan requirements]

4. In aircraft configuration the search process is mainly guided by two types of goals in accordance with the airworthiness requirements for the type of aircraft. First, choices are made in order to satisfy the specification and success is judged by how close the performance comes to that required. Second, choices are made in order to maximise or minimise various parameters e.g., range, payload, fuel consumption, weight etc. Thus, the use of heuristics to guide the search process and to reduce the amount of computation for example,

   [USE turbofan
   BEFORE propfan]

   [USE wing mounted engines
   BEFORE fuselage mounted engines]

   [USE two engines
   BEFORE three engines]

5.2.5 Consequences

A large part of aircraft configuration knowledge involves deducing how a single action will lead to changes in others. For example,

ACTION: choose a high wing

CONSEQUENCE: provides ground clearance
floor level is low
good outside view for crew and passengers
accessibility to engines is difficult
accessibility to wings is difficult
decreases ground effect

ACTION: choose nosewheel undercarriage
CONSEQUENCE: heavy braking cannot cause nosing over
brake drag forces act behind the cg
initial take-off attitude has a low drag
nose down pitch resulting from a two point landing
pilot's view is relatively good during taxying
on the ground the fuselage is roughly horizontal
relatively high weight of the nosewheel
may need tail bumper or stiffened rear fuselage
horizontal attitude of the aircraft assists braking
main wheels may be difficult to mount
nosewheel may be difficult to mount
retraction of the nosewheel can be difficult

If the above actions succeed its consequences will lead to further actions and consequences. For example,

ACTION: low floor
CONSEQUENCE: quick loading/unloading of passengers
quick loading/unloading of cargo
provides good access to the fuselage

ACTION: ground effect
CONSEQUENCE: reduces take-off distance

increases landing distance

From the above examples, it can be observed that the knowledge involved is qualitative i.e., it gives the sign of the change but not the amount. This sort of knowledge is useful in working out possible design solutions which can then be evaluated more accurately. Another use is to allow improvement of an existing design by knowing the consequences of various actions.

5.3 CONCEPTUAL DIFFERENCES

Based on the knowledge present in aircraft configuration and wing design the following conceptual differences can be identified:

Subproblem interaction. Wing design is a very precise process within aircraft design, which to a large extent does not depend on other design steps. Thus, the knowledge built into the wing design process is highly mathematical in that for the most part it uses well defined formulae to evaluate different parameters and it uses factual information to represent the aerodynamic data corresponding to each aerofoil section used. In addition the flow of knowledge through the system and the consequent ordering of tasks is highly sequential. As a result the interaction between tasks is limited and the required inferencing procedures are relatively uncomplicated. Aircraft configuration on the other hand is less rigid and relates more to the designer's experience and resources in order to compare and judge different configurations according to an aircraft specification. A designer when considering any one of the aircraft main components must constantly refer to the other components in order to preserve the overall consistency of the solution.

Design procedure. In many cases during aircraft configuration there is no single answer to a problem and there is no ideal method to follow for every situation. The nature of the configuration module makes the design process look non-deterministic in that the expert does not follow any well defined path towards an optimum solution. In reality a pool of knowledge is used in arriving at a solution which may not be the optimum solution but which best matches the specification. Within this pool of knowledge the system will need to know the 'knock-on' effects of decisions made early on in the design in order to weight the different factors involved and pass some kind of judgement towards a suitable solution. This is in contrast to the wing design problem which can be solved as a sequence of well defined and independent steps.

Re-design in the configuration mode represents a more elaborate task than that encountered in wing design. When the design fails, perhaps because of a bad decision made earlier, the designer backtracks in a very clever and precise way. He is guided both by the knowledge accumulated during the current design process and by experience built up during earlier design exercises. The physical properties of the structure, aerodynamic loads etc., constrain the choices available and must be taken into account. In addition, the designer takes into account the fact that changing a design in order to correct a fault or error must not be done in such a way as to introduce problems later in the design process. Thus, the process of moving from an inadequate solution requires a complex interaction of the design history knowledge accumulated during the design process and new knowledge introduced to guide the designer to a new solution.

Design history. Because of the size of the configuration module, the design process may go on for several days at a time thus the designer will find it difficult to remember all previous designs. The system will be required to remember its own past history i.e., know past solutions to inform the designer of things which he has done before in order to avoid a bad decision which has been previously tried. Thus, keeping track of the design history helps a designer improve the design method used, gaining a better understanding of his own decision process.

5.4 EXTENSIONS TO THE BASIC APPROACH

The above conceptual differences between aircraft configuration and wing design are used in this section to describe the extensions necessary to the aircraft design expert system framework defined in Chapter 4 in order to achieve a suitable implementation of the aircraft configuration module.

5.4.1 User interface

Because the process is complex, more effective methods for interacting with the user are required employing an engineering oriented interface. This may involve interpreting a parametric description of the design in terms of images or engineering jargon. Graphics can be regarded as the most natural user language for the designer using a computer and represents one of the areas where the greatest flair is required in order to achieve the high level of novelty and clarity necessary. The graphical display of different design components that are reasonably accurate with respect to the dimensions is indispensable in the design process. The designer usually likes to see how the design components fit together in order to judge if the design 'looks right' thereby detecting early on, a design which might not work further down stream. Examples of this visual aid to the designer include:

1. Engine dimensions are often of particular importance in view of their relation to duct sizes, landing gear height etc.

2. Display the way the main undercarriage retracts and fits under the wing in order to check obstructions.

One of the problems with any large design is that of being unable to guide the designer through the different design steps easily. Some of the questions which need to be answered, but in a much more explicit and complete way than that required by the wing design module are:

1. Where am I in the design process?

2. How much of the wing design have I done?

3. How many more steps to be performed?

There exists different methods in which the system could display the evaluation of the design steps. A widely used and relatively simple method to implement is by displaying the design as a tree in which each node represents a design step and is connected to other nodes by lines. This representation may be appropriate in certain cases but if used with engineers the representation could be difficult to follow and even discourage the designer from using the system. Thus, there is consideration to be given to alternative methods of displaying the design process. An approach with configuration may be the drawing of the different parts with respect to each other as they are chosen.

5.4.2 Templates

The template concept described in Chapter 4 has been developed to enable the representation of design knowledge as a series of small and independent design steps. Thus, at first, the template concept developed for wing design seems appropriate to aircraft configuration as the description has been based on selecting, sizing, and positioning the aircraft main components such that these steps may be represented as a sequence of control and action templates. But, due to the complex interaction between the different design steps and the inability to sequence these steps makes the template representation unsuitable for all but the simplest (i.e., well understood and independent) tasks. Thus, there is a clear requirement for a scheme which will effectively represent the aircraft configuration knowledge allowing the interaction between the different design steps and the deduction of further knowledge without making assumptions about the sequence of events.

5.4.3 Global controller

The template (or goal) representation of the wing design process presented in Chapter 4 captures the subgoal-supergoal, conjuctive, and sequential relationships among the different design steps as seen in Figure 4.2 but fails to capture interacting goals i.e.,

1. Transformations used to achieve a goal may affect multiple portions of the design tree

2. A goal must be achieved before another goal in a different part of the design tree

3. Achieving a subgoal helps achieve a goal other than its ancestors in the goal tree

4. Two goals cannot both be achieved

Error correcting in aircraft configuration as described in section 5.3, represents both a forward and backward looking process with a strong interaction between design rules and calculations.

Futhermore, the system needs to know the knock-on effects of decisions made early on in the design. For example, in selecting a wing mounted undercarriage for a high wing aircraft, the main undercarriage will be long and heavy, increasing fuel consumption which will require more fuel to be carried to achieve the same operating range thereby requiring bigger fuel tanks, etc. Whereas having attached the undercarriage under the fuselage would have outweighed the disadvantages.

As a result of the above considerations, a configuration controller needs to handle these complex tasks whilst considering the interactions between the different abstraction levels, particularly during a backward or forward looking procedure. This requires the ability to handle multi-task procedures where various parts of the design interact.

5.4.4 Design representation

When an engine layout is specified by the designer an error may occur later in the design process due to this choice. The system is to take into account the range of choices available without actually using them during a particular design cycle, and make comments or suggestions as to the possible actions to take in order to correct the design error. Futhermore, a one to one matching of the database elements to the requirement will not usually occur and hence no single aircraft can be used as the nearest to the current need. This matching is likely to be satisfied by various parts of different aircraft each having its pros- and cons- towards the required aircraft. Thus, the system will need to weight the pros- and cons- towards appropriate decisions and be able to correctly interprete the concepts of nearest at each stage of the matching process.

The specification points towards a particular aircraft in the database of previous designs, parts, layouts etc., with similar specification. In this case the requirement consists of a knowledge based guidance system to allow the system to move through the design database elements to appropriate points.

5.5 CONCLUSIONS

The present Chapter has been concerned with aircraft configuration by identifying the knowledge used by the designer. Various examples of the types of knowledge present in aircraft configuration have been given, and the conceptual differences with wing design identified. Based on this analysis, the necessary extensions to the framework proposed in Chapter 4 for an aircraft design expert system have been described.

The next Chapter describes a computer implementation of the wing design problem using the aircraft expert system framework proposed in Chapter 4.

CHAPTER 6

TRIAL IMPLEMENTATION

6.1 INTRODUCTION

The objective of the research presented in this thesis is aimed at the development of an expert system for aircraft design. However, the development of a complete system is beyond the scope of the present work so as a first pass towards this goal, a prototype computer program for wing design was developed to trial concepts for building an aircraft design expert system. In the preceding Chapters the significance of the wing design task within aircraft design has been discussed and the fundamental knowledge representation issues identified.

This prototype is called ADROIT (Aircraft Design by Regulation of Independent Tasks) and, from within the total wing design, it designs a high subsonic aircraft wing by selecting a two-dimensional aerofoil section from a choice of three alternatives and evaluates a range of suitable sweep angles according to the aircraft specification. A controller monitors the execution of the different design steps and allows the designer to re-evaluate any of the previous steps if one fails. Great emphasis has been made on making ADROIT easy to use.

The data input is free-format and it is performed interactively from the terminal. The system contains a set of error diagnostics which advise the user of wrong input and failure of a design step. The data output is clear and self explanatory enabling the user to interpret the results without delay. A set of commands is available to the user to control the extent and form of the consultation.

The design techniques used (1) in the program have been implemented and tested at Cranfield, and employed by aircraft design students and lecturers to generate and check various wing configurations. Futhermore, configurations for which the system was not originally designed for have been successfully performed e.g., low subsonic civil aircraft.

The system is supported by a User's Manual (27) which describes the overall capabilities, summarises the theories implemented, basic input data format, and the command set available to the user during a consultation. An accompanying Programmer's Manual (26) describes the program structure layout and the procedures for extending and modifying the code. The program has been implemented in the logic based PROLOG language and runs on the following computers and operating systems: VAX-11 750/VMS, SUN 3/UNIX, and IBM PC/MS-DOS. From among these, the latter implementation is described here, as it is felt that this offers the greatest flexibility to potential users.

Chapter 4 defined a six module framework for an aircraft wing design expert system, and outlined the status of each module within the present implementation together with the knowledge representation aspects and techniques used. The current Chapter draws on this earlier Chapter in order to describe the scope, procedures used, operating instructions, and presents a record of a consultation session with ADROIT.

6.2 DESIGN PROCEDURE

Figure 6.1 shows a flow-chart of the wing design process as implemented within the ADROIT program following the solution methodology used by the expert designer. Notice that this representation does not show the many iteractions which can occur at any stage during the design process but which are taken into account by the ADROIT global controller as described in Chapter 4. Because of the wide range of aircraft types and solution techniques available, the following assumptions have been made in order to restrict the scope of the design:

1. The types of aircraft considered will be limited to high subsonic passenger or cargo civil transport aircraft.

2. Three supercritical aerofoil sections suitable for high subsonic speeds (Mach numbers between 0.7 and 1.0) are used. Although, it is possible to insert other type of sections to cater for slower aircraft.

3. The mass and aeroelastic procedures assume conventional aluminium alloy construction. Subsequent versions of ADROIT could cater for the use of composite materials. The program, however, does allow for the use of active controls.

4. ADROIT designs a wing for cruise conditions, but allowance is made for take-off and landing requirements.

Furthermore, the design procedures described below assume that the aircraft specification and the parametric study outlined in Chapter 3 have been performed.

6.2.1 1st STAGE: two-dimensional wing design

As described in Chapter 4, the first choice to be made concerns the selection of an efficient aerofoil shape based primarily on the cruise Mach number and the cruise lift coefficient. In this area theoretical predictions are difficult thus, the knowledge base stores wind tunnel results of 'real' aerofoils as described in Chapter 4. There is in ADROIT a choice to be made between three supercritical sections (75), two at 10.5% t/c (RAE 9515 and 9530), and one at 12.2% t/c (RAE 9550). In choosing the overall best section the program uses the cruise Mach number and the cruise lift coefficient defined by the user to compare these sections and selects a 'best' section in terms of maximum lift coefficient and stall behaviour at low speeds, and 2-D drag rise, lift/drag ratio, and pitching moment at cruise conditions.

The term 'relative importance' is used in the ADROIT program to indicate the importance of a specific parameter employed in generating an acceptable wing design. In effect an expert supplied 'weighting factor' is used to bias values of parameters or parameter differences in order to produce the correct rating between these parameters. Figure 6.2 shows in a tabular form the relative importance attributed to each parameter according to a relative difference e.g., when rating a section stall lift coefficient the difference between this section and the worst is calculated, say it is equal to 0.2 then a rating of 10 is given to the former section. For each section, the parameter ratings are summed and the section with the highest rating corresponds to the overall best section. Finally, note that the higher importances attributed to the lift/drag ratio and the 2-D drag rise when compared to the other parameters is due to the desire in obtaining an efficient high subsonic wing section.

6.2.1.1 Lift coefficient - at low speeds and high incidences should be high in order to avoid early stall. The maximum lift coefficient at low Mach numbers for each section are stored in the database. ADROIT retrieves these values and rates each section using:

$$i = (ri/rd) * d = (10/0.2) * d \qquad (1)$$

Where i is the importance of this parameter towards the selection of the best section, ri and rd are the relative importance and relative difference obtained from Figure 6.2, and d is the difference in stall lift coefficient between the current section and the worst section. This procedure is repeated for each of the parameters described below.

6.2.1.2 Stall behaviour - refers to how sudden the drop in lift coefficient is after the stall. In order to measure this effect, the slope over one degree of incidence after the stall at low speeds has been measured for each section. ADROIT retrieves these values from the database and rates each section using:

$$i = (ri/rd) * d = (8/0.5) * d \qquad (2)$$

6.2.1.3 Cruise lift/drag ratio - should be as high as possible i.e., minimum drag coefficient. At the specified economic cruise Mach number and cruise lift coefficient the drag coefficient is found for each section by interpolating or extrapolating the wing section data as required and the program rates each section using:

$$i = (ri/rd) * d = (15/0.005) * d \tag{3}$$

Notice that the actual lift/drag ratio is not used as the cruise lift coefficient is a constant in the computations i.e., it acts as a normalising factor between the sections.

6.2.1.4 Pitching moment - should be of low to moderate magnitude at the specified economic cruise Mach number and cruise lift coefficient. The pitching moment is found for each section by interpolating or extrapolating the wing section data as required, and the program rates each section using:

$$i = (ri/rd) * d = (10/0.05) * d \tag{4}$$

6.2.1.5 2-D drag rise - refers to an increase in drag due to the compressibility effects and at the specified cruise lift coefficient the critical Mach number should be sufficiently high to avoid this effect. From the stored wing section data, ADROIT evaluates the drag rise Mach number for each section and rates them using:

$$i = (ri/rd) * d = (30/0.1) * d \tag{5}$$

6.2.2 2nd STAGE: three-dimensional effects

It is necessary to achieve an acceptable design in terms of 3-D drag rise, aeroelastics, tip stall, flap effectiveness and weight. This is done by proposing a range of sweep angles at quarter chord between 15 and 45 degrees in steps of 5 degrees. The appropriate t/c ratios are determined for suitable drag rise characteristics and the other properties checked and unsuitable sweep angles eliminated in a sequence similar to that shown in Figure 6.1. The assesment methods for the aerodynamic and structural properties are given below.

6.2.2.1 3-D drag rise - should be acceptable for a given sweep angle. This is an iterative process using design equations to select a sweep angle and t/c ratio which gives the required low drag at the aircraft economic cruise Mach number (Mecon). The 3-D drag rise Mach number (MD) is dependent upon the wing planform and aircraft layout as described in Chapter 3. In the case of an unswept wing the aspect and taper ratios have little effect providing they are not unusually low. However, the presence of a fuselage or nacelles does reduce MD unless special precautions, such as area ruling, are taken. A normal body reduces MD by between 0.02 and 0.05 (1) depending upon the rate of growth of body cross-section in way of the wing, or in other words the local increment given to Mach number by the body relative to the free stream value. Fuselage effects are included in ADROIT using semi-empirical rules to account for wing-fuselage interaction. The user defines the wing-fuselage interaction effect guided by the following aircraft:

BAe 125	(0.05)

Boeing 747	(0.035)	(6)
Boeing 727	(0.02)	

The values in brakets (not shown to the user) represent the reduction (Decr) in 3-D drag rise Mach number according to the configuration choosen. The 3-D drag rise Mach number at $\Lambda = 0$ is evaluated using:

$$(MD)_{\Lambda=0} = Mecon + Decr \tag{7}$$

For a finite wing, the 3-D drag rise at a given sweep back angle is (1):

$$MD = (MD)_{\Lambda=0} / \cos^{1/2}\Lambda_{1/4} \tag{8}$$

The following relation is used to generate a section (i.e., different t/c) for a given sweep angle:

$$t/c = (t/c)1 \frac{(MD - MD2)}{(MD1 - MD2)} + (t/c)2 \frac{(MD - MD1)}{(MD2 - MD1)} \tag{9}$$

Where subscripts 1 and 2 denote the best two sections with different t/c. This equation is derived using a linear relation between these sections. Note that as ADROIT stores only two sections with different t/c (i.e, 10.5% and 12.2%) innaccuracies occur when generating results outside this range.

The program checks that the high-speed requirement can be met by evaluating the difference between the 3-D drag rise Mach number and the maximum cruise Mach number (Mmax) defined by the user. This difference should not be greater than 0.02 i.e.,

$$M_{max} - M_D \leq 0.02 \quad (10)$$

A further check is performed in order to eliminate those sweep angles which have generated sections with t/c greater than 18% as these sections are unaceptable for the type of design being considered.

6.2.2.2 Aeroelastic stiffness - The torsion and bending stiffness of the wing are checked to see if sweep and t/c are satisfactory from the aeroelastic viewpoint. These checks are carried out using elementary formulae (1) derived from empirical data on similar aircraft.

The user defines the engine position, whether or not active controls are to be used, the cruise altitude, and the initial aspect ratio. The expression used (1) to check torsional stiffness with wing mounted engines is:

$$\frac{A^{3/2}}{(t/c)^2} < \frac{3 \times 10^8}{V_D^2 \cos \Lambda_{1/4}} \quad (11a)$$

For non-wing mounted engines:

$$\frac{A^{3/2}}{(t/c)^2} < \frac{2.5 \times 10^8}{V_D^2 \cos \Lambda_{1/4}} \quad (11b)$$

The diving speed (VD) for a given altitude (i.e., speed of sound) is computed using the following expression (1):

$$V_D = (M_{max} + 0.05)\, V_{sound} \quad (12)$$

The following expression is used (1) to check the bending stiffness:

$$\frac{A^{3/2} \sec \Lambda_{1/4}}{(t/c)} < \frac{850}{N} \tag{13}$$

Where N is the ultimate load factor and taken to be 3.75 for large transport aircraft and is reduced to 2.5 when using active controls.

In order to obtain a satisfactory aeroelastic performance scaling is performed by decreasing the initial aspect ratio by up to 10%. If complex changes are required then the user requires prompting with respect to possible changes to the basic design. The overall valid range of sweep angles corresponds to the aspect ratios which have satisfied both the torsion and bending stiffness checks.

6.2.2.3 Tip stall - is performed by evaluating the spanwise airload distribution using a combination of Schrenk and Stanton-Jones formulae (1) to take into account of sweep back, wing twist, and taper ratio. The effect of camber or local aerofoil section changes are not included. The following expressions are combined to evaluate the spanwise lift coefficient distribution CL(y):

Basic load distribution using Schrenk's approximate method:

$$\frac{CL(y)/\overline{CL}}{c(y)/\bar{c}} = \frac{K\, a_0(\alpha_0 + \epsilon)\ /\ \overline{CL}}{3\ (\eta\lambda^2 - \eta + 1 + \lambda)\ /\ 2\ (\lambda^2 + \lambda + 1)} \tag{14}$$

Additional load distribution using Stanton-Jones formula:

For $\eta < 0.7$:

$$\frac{CL(y)/\overline{CL}}{c(y)/\bar{c}} = 1.28(1 - \eta^2)^{1/2} + (14.13\eta - 6.35)(\bar{y} - 0.425) \tag{15a}$$

For $\eta >= 0.7$:

$$\frac{CL(y)/\overline{CL}}{c(y)/\overline{c}} = 1.28(1-\eta^2)^{1/2} + [4.25 - 53.8(\eta - 0.815)^2](\bar{y} - 0.425) \quad (15b)$$

Where $\bar{y}$ is the spanwise position of the centre of pressure and is given by:

$$\bar{y} = 0.42 + A\, m\, [(4.4 + 5\lambda) \tan\Lambda 1/4\, /m + 10.4\,\lambda^{1/2} - 6.7]/10^3 \quad (16)$$

The user defines the taper ratio and the wing twist which is assumed to be linear i.e., zero degrees at the root chord and 'n' degrees at the tip chord (wash out). For the range of sweep angles which satisfied the aeroelastic stiffness check, the spanwise local lift coefficient distribution is evaluated using the above expressions. If the local lift coefficient along the span exceeds the cruise stall lift coefficient (i.e., the stall lift coefficient at the 2-D drag rise for the best section) then, the sweep angle at which it occurs is eliminated. If a stall condition is predicted then the taper ratio must be changed or more pre-twist introduced by the user.

6.2.2.4 <u>Flap effectiveness</u> - is reduced by increasing wing sweep. The maximum landing lift coefficient (CLmax) defined by the user is compared against the effective landing lift coefficient (CL cos Λ1/4) assuming leading-edge slat and double-slotted flaps and allowing 1.15 CL for basic wing and 0.5 CL for leading edge devices (1).

$$CLmax =< CL * \cos \Lambda 1/4 \quad (17)$$

Where CL = 3 for the type of flaps used (2), and the above inequality is evaluated for a range of sweep angles eliminating those for which it is not satisfied.

6.2.2.5 Wing weight - Because the ADROIT program represents only part of the complete design process, it does not know the total aircraft mass. To overcome this deficiency the program requests from the user the all up mass of the aircraft and the wing loading together with the operating range. With this information the approximate wing weight is evaluated for different combinations of sweep angle, t/c, and aspect ratio using the following expression (1):

$$Ww = C \left[\frac{bS}{\cos\Lambda 1/4} \left(\frac{1 + 2\lambda}{3 + 3\lambda}\right) \left(\frac{WN}{S}\right)^{0.3} \left(\frac{VD}{t/c}\right)^{0.5} \right]^{0.9} \qquad (18)$$

Where C is a constant which depends on the aircraft type, and is obtained using the following relation:

$$C = C1\ (R - R1)/(R1 - R2) + C2\ (R - R2)/(R2 - R1) \qquad (19)$$

Where C1 = 0.034 and R1 = 9000 nm for long range aircraft, C2 = 0.028 and R2 = 500 nm for short range aircraft, and R is the user defined operating range. Although this design step is not a check on sweep angle, it is useful as a comparison measure between different sweep angles as the designer aims at the lightest solution which has satisfied the above checks.

6.2.3 Iteration

The overall valid range of sweep angles is obtained by combining the results from the aeroelastic, tip stall, and flap effectiveness checks and are presented to the designer together with the corresponding aspect ratio, t/c, and wing weight to allow the optimum sweep angle to be selected. If an overall valid sweep angle is not found, the designer can re-evaluate one or more of the previous design steps in order to arrive at an optimum range of sweep angles. For example, active controls may be considered in order to reduce the wing bending moment and hence the effective load factor. (The wing planform, design for take-off and landing, and final sizing as described in Chapter 3 are not considered by ADROIT).

6.3 OPERATING INSTRUTIONS

ADROIT uses various windows to provide a powerful man-machine interface. It requests input using a variety of menus and screens, displays data and results in a text or graphical form, provides help at any stage during a consultation session, advices the user of errors in the design, etc. This section describes the system operating instructions.

6.3.1 Accessing ADROIT

ADROIT is supplied in a floppy disk from which it can be run by simply typing ADROIT and pressing Return or from a hard disk. But, it is best run from the latter as the image file is quite large (the executable image being 250 Kbytes). The files on the distribution disk are:

1. ADROIT.EXE - contains the executable image of the system and it is the only file necessary to run ADROIT.

2. NAME.TXT - contains the introduction text to the system and to each design step. When NAME is associated with a design step name, it is shorten to eight characters as MS-DOS is unable to handle longer file names.

3. NAME.MEA - contains the definition of a question together with the valid range of answers. NAME denotes the symbol for the parameter.

4. NAME.EXA - contains the example answers to various input questions. NAME denotes the symbol for the question being asked.

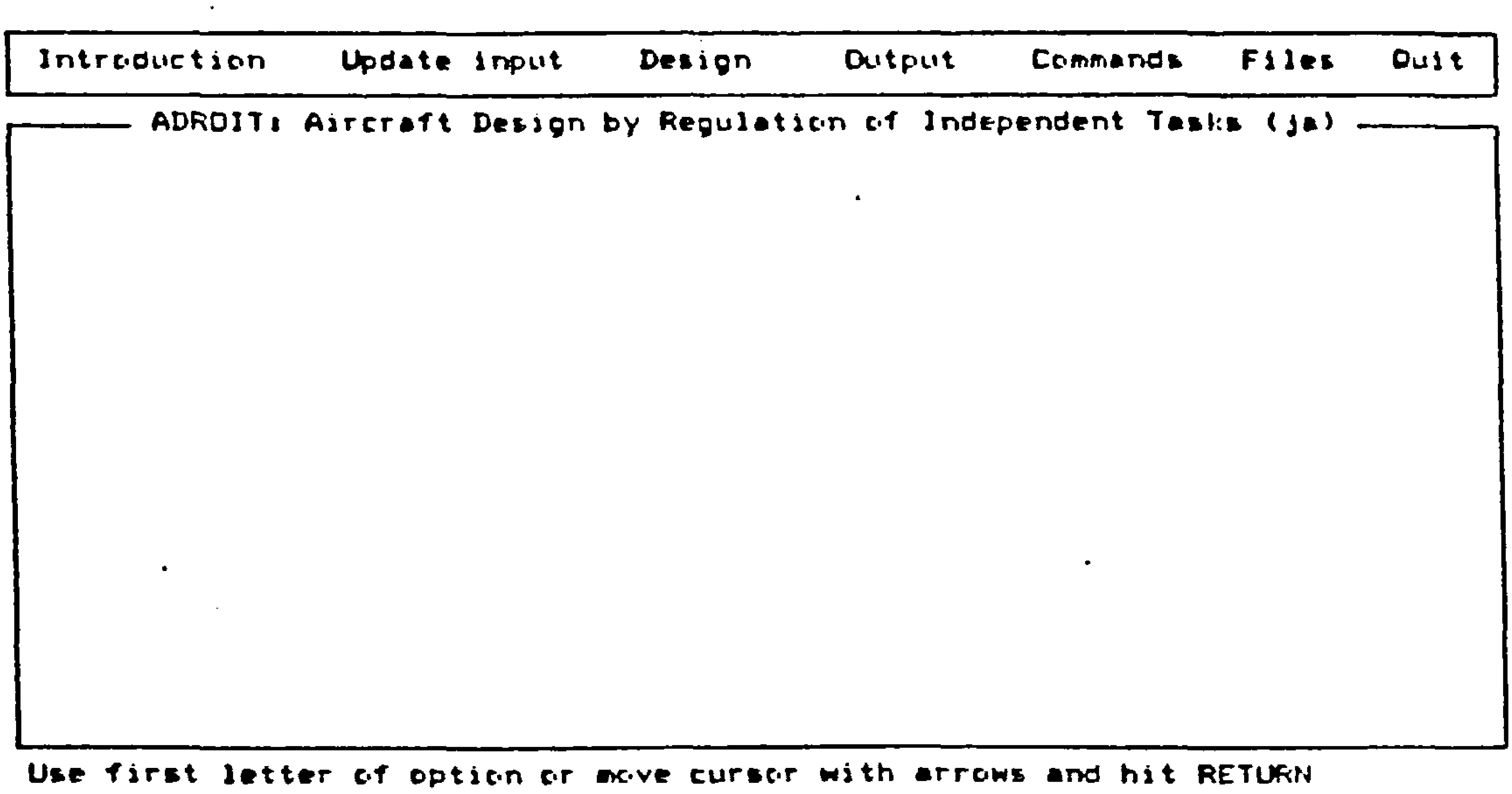
During a consultation session two types of files are created:

1. NAME.OUT - stores the results after evaluation (or re-evaluation) of a design step NAME, together with the file full name, directory, date and time of creation.

2. NAME.SAV - is created whenever the user saves the state of the design under NAME.

6.3.2 Main menu

The ADROIT main menu is a pull down menu located at the top of the screen and displays seven options as shown below.



6.3.2.1 Introduction menu - displays information about the system or any design step.

6.3.2.2 Update input command - displays a screen with all the questions and default (or previous) answers required by the design steps as shown below.

```
                         WING DESIGN INPUT DATA
                         **********************

Nominal Operating Range (nm)    3000    Optimum Cruise Altitude (ft)   38000

Maximum Cruise Mach Number      0.78    Economic Cruise Mach Number    0.75

Cruise Lift Coefficient         0.57    Landing Lift Coefficient       2.6

Wing Aspect Ratio               9.9     Wing Taper Ratio               0.3

Wing Twist (deg)                3       Maximum Wing Loading (Kg/m2)   500

All Up Mass (Kg)                53000

Fuselage Wing Interaction       Underwing Pod Engines      Active Controls
BAe 125                         yes                        no

Fill in details. Move cursor with arrows. RETURN:Select F1:Hlp F10:End ESC:Quit
```

6.3.2.3 Design menu - presents the user with a menu showing the possible design steps to perform according to the design state.

6.3.2.3.1 Selecting a design step - is required in order to carry on with the design evaluation. A menu is offered to the user which includes the step used to go back up the design tree described in Chapter 4. When the exit option is offered, it can be used to return to the main menu, a similar effect is achieved by pressing Esc continuosly. After selecting a design step, the system displays any sub-steps for further selection.

6.3.2.3.2 Viewing a design step - depends on whether the step is represented as a control or as an action template.

Control: NAME offers a two option menu to view the control template NAME in terms of: an introduction option to introduce the step and an output option to display the results corresponding to the templates which constitute the control template.

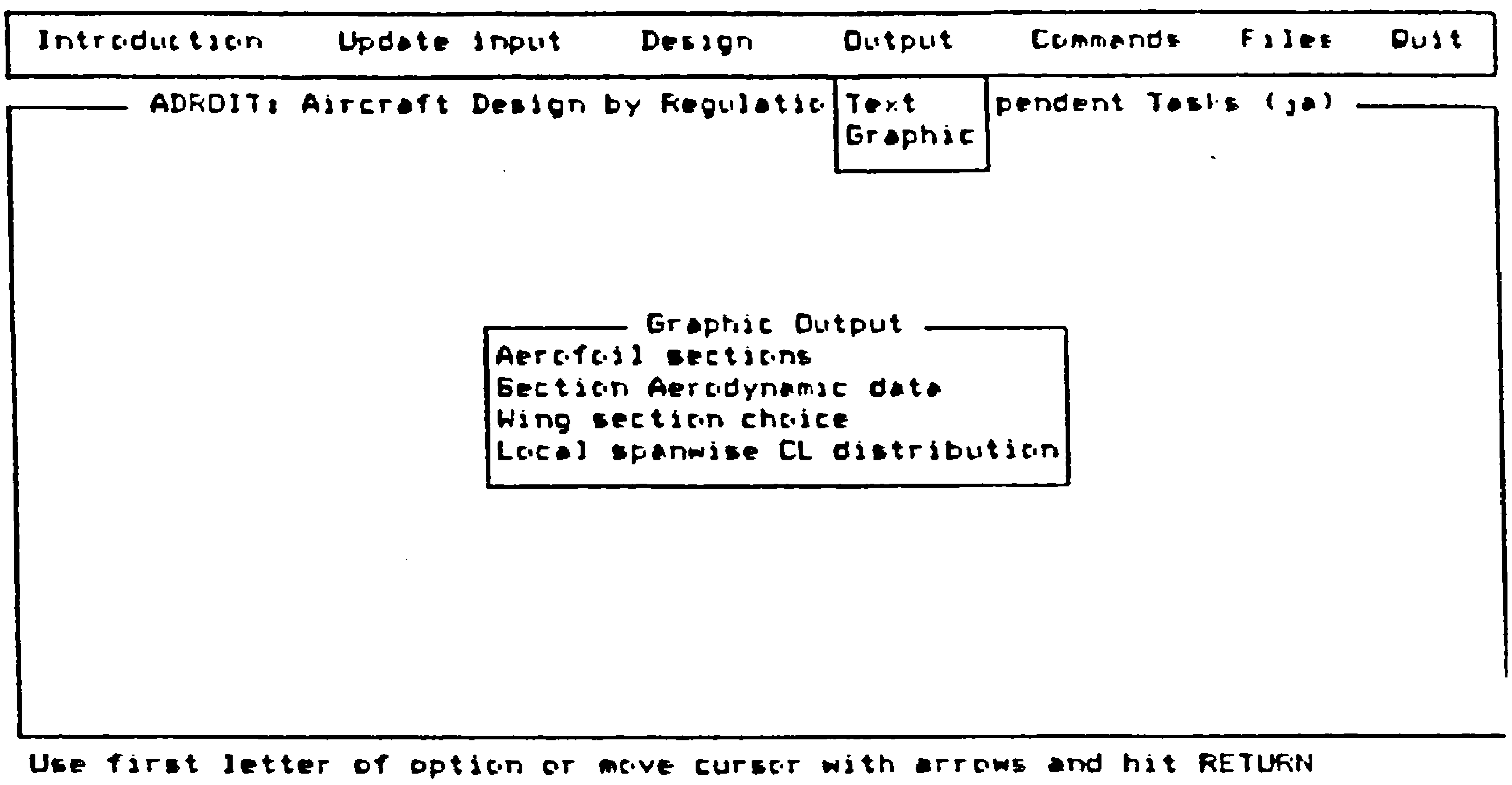
Action: NAME offers a three option menu to view an action template NAME in terms of: an introduction option to introduce the step, an Update input and re-evaluate option to change input specific to the step, and an output option to display the results evaluated by the step.

6.3.2.3.3 Executing a design step - If the design step selected is an action template the system executes the design step and displays the message 'Doing: NAME' on the message window. When it has been completed, the results are written to the file NAME.OUT and the system offers the action template menu described above.

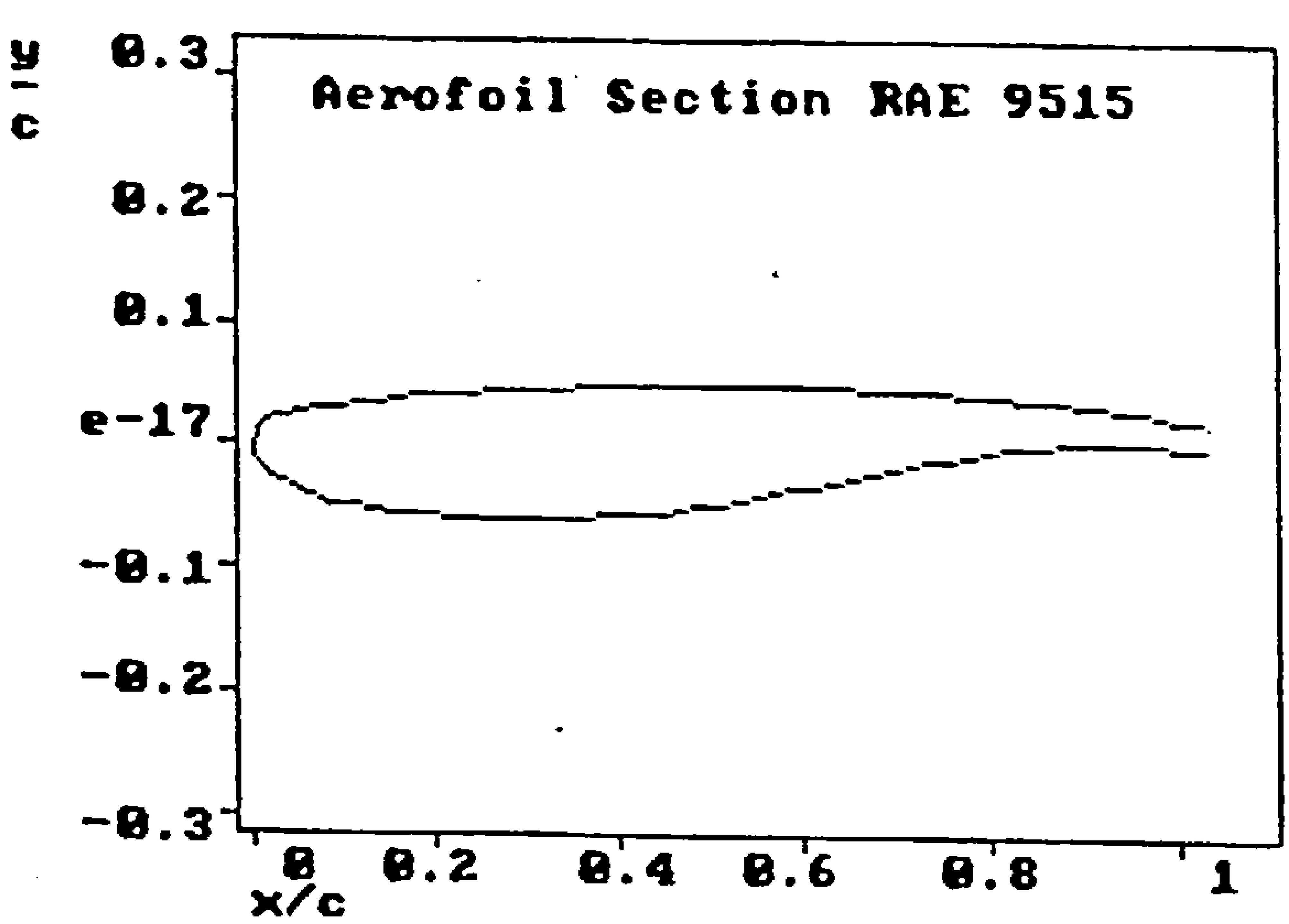
6.3.2.3.4 Validity of a design step - When the user exits a template menu the system displays the message 'Evaluating: NAME' on the message window and checks the results. If the results are found to be unsatisfactory by the system it displays the cause of failure on the error window and it offers a menu with the design steps which could have caused the design failure. The user can select one or more design steps for re-evaluation.

6.3.2.3.5 Finalizing the wing design - When all the design steps have been evaluated the system indicates in the message window that the design has been completed and returns to the main menu.

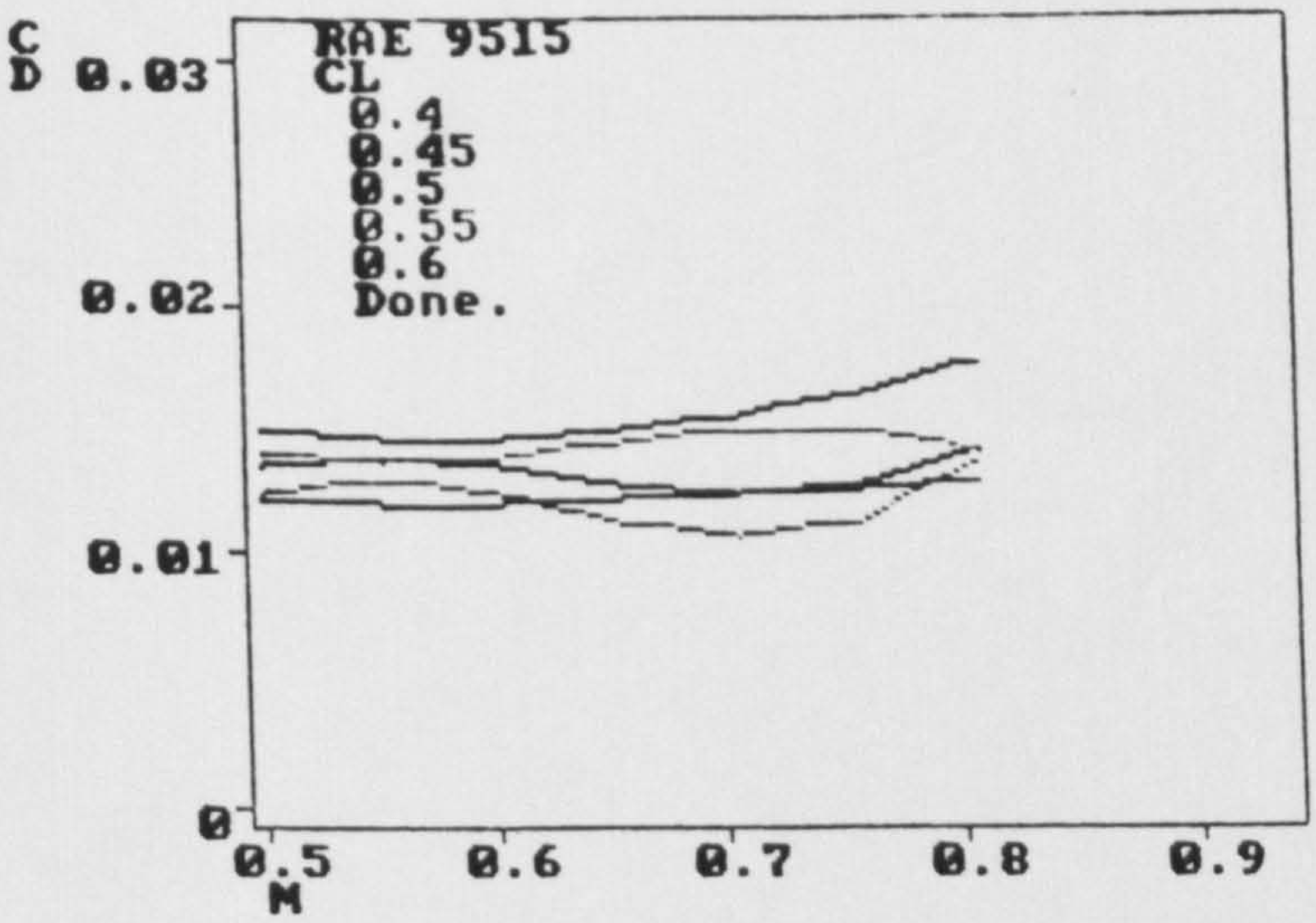
6.3.2.4 Output menu - displays a two option menu. The Text option is used to view the results for the design steps (action and control templates) which have been evaluated. The Graphics option offers the four option menu shown below.



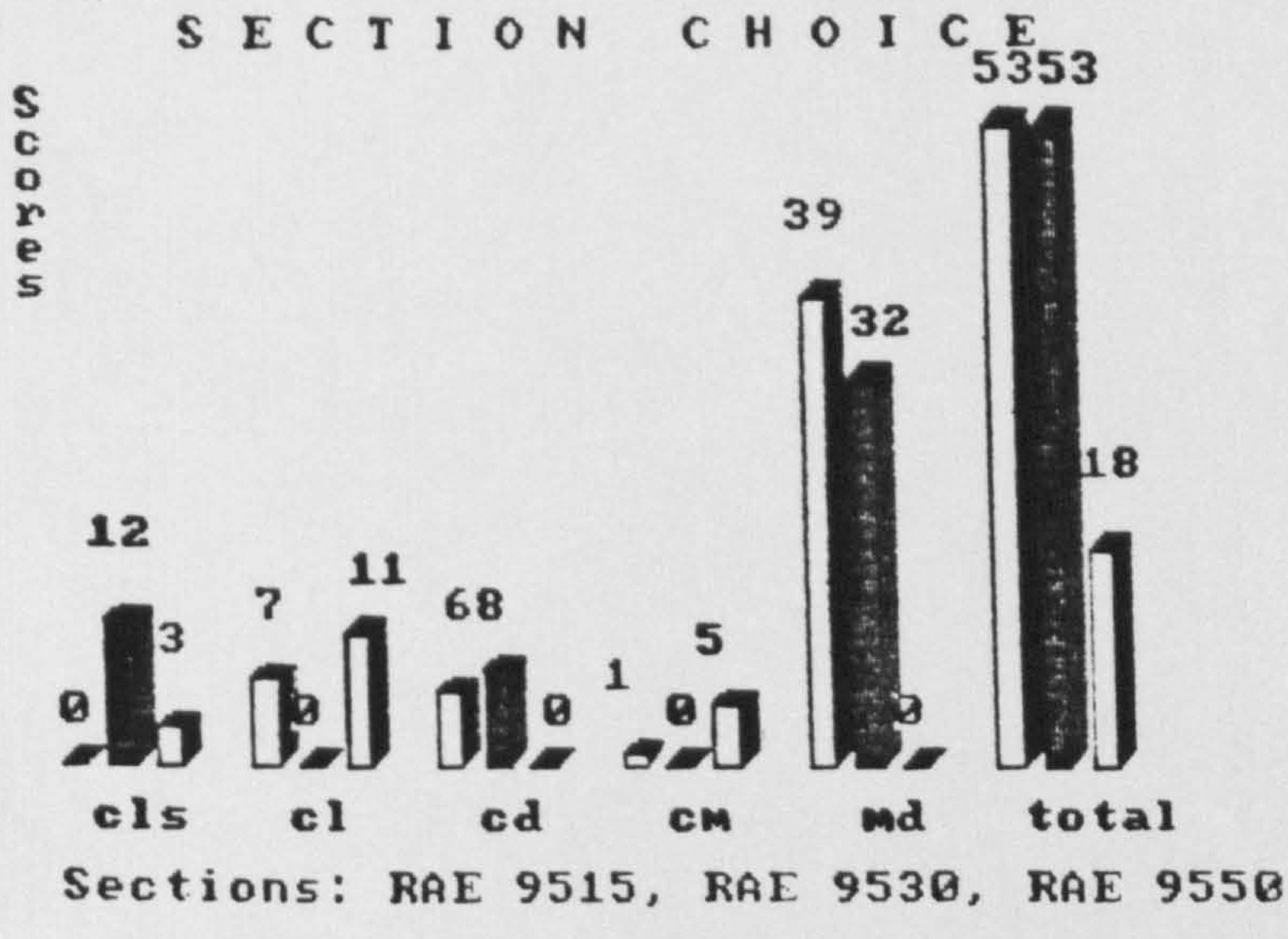
The Aerofoil sections option displays the three supercritical aerofoil sections used of which Section RAE 9515 is shown.



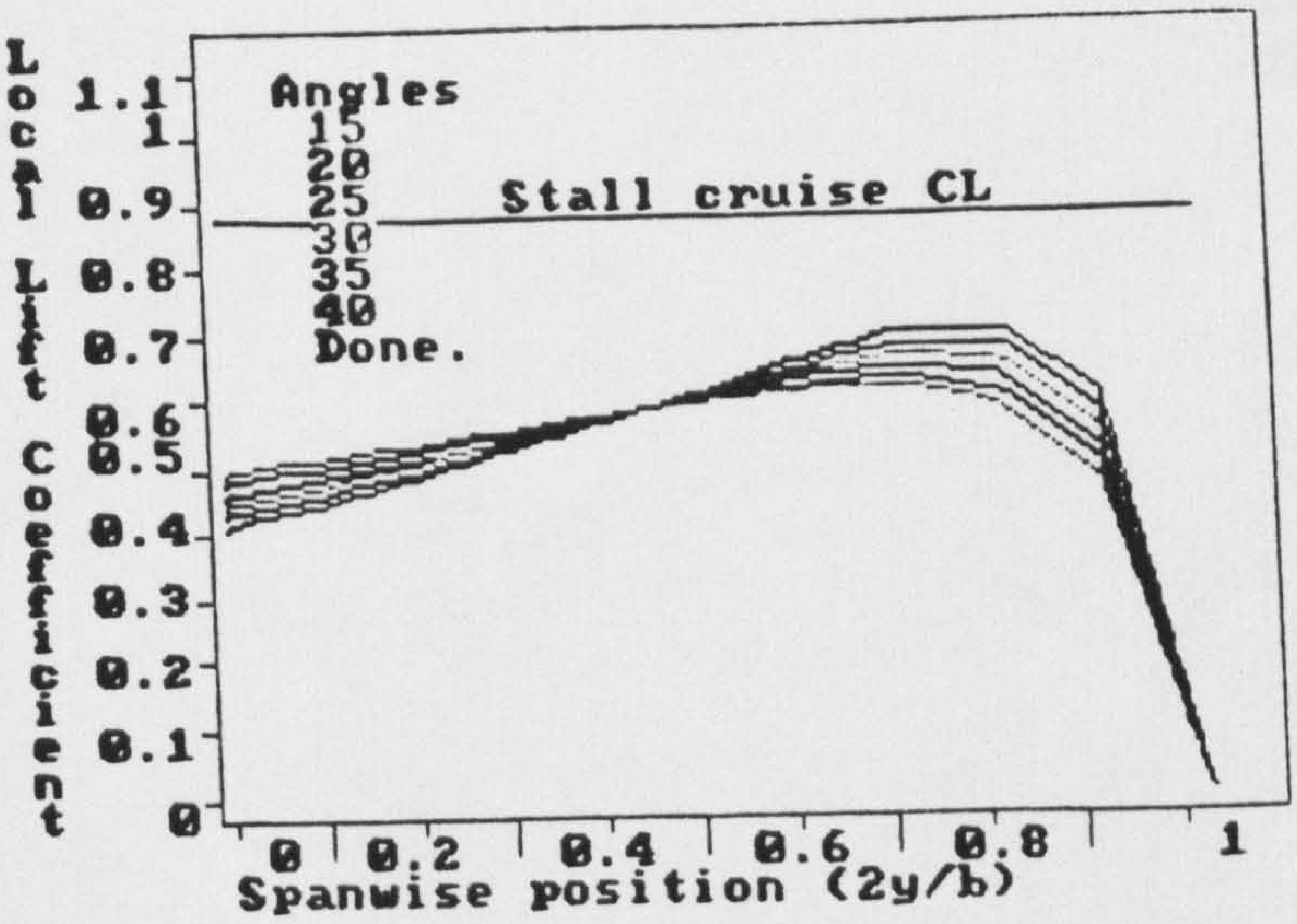
The Section aerodynamic data menu displays the drag coefficient, pitching moment and two-dimensional drag rise for each section. The drag coefficient for section RAE 9515 is given below.



The scores for the various wing sections are compared on a bar chart of the section ratings together with their total ratings as shown below.



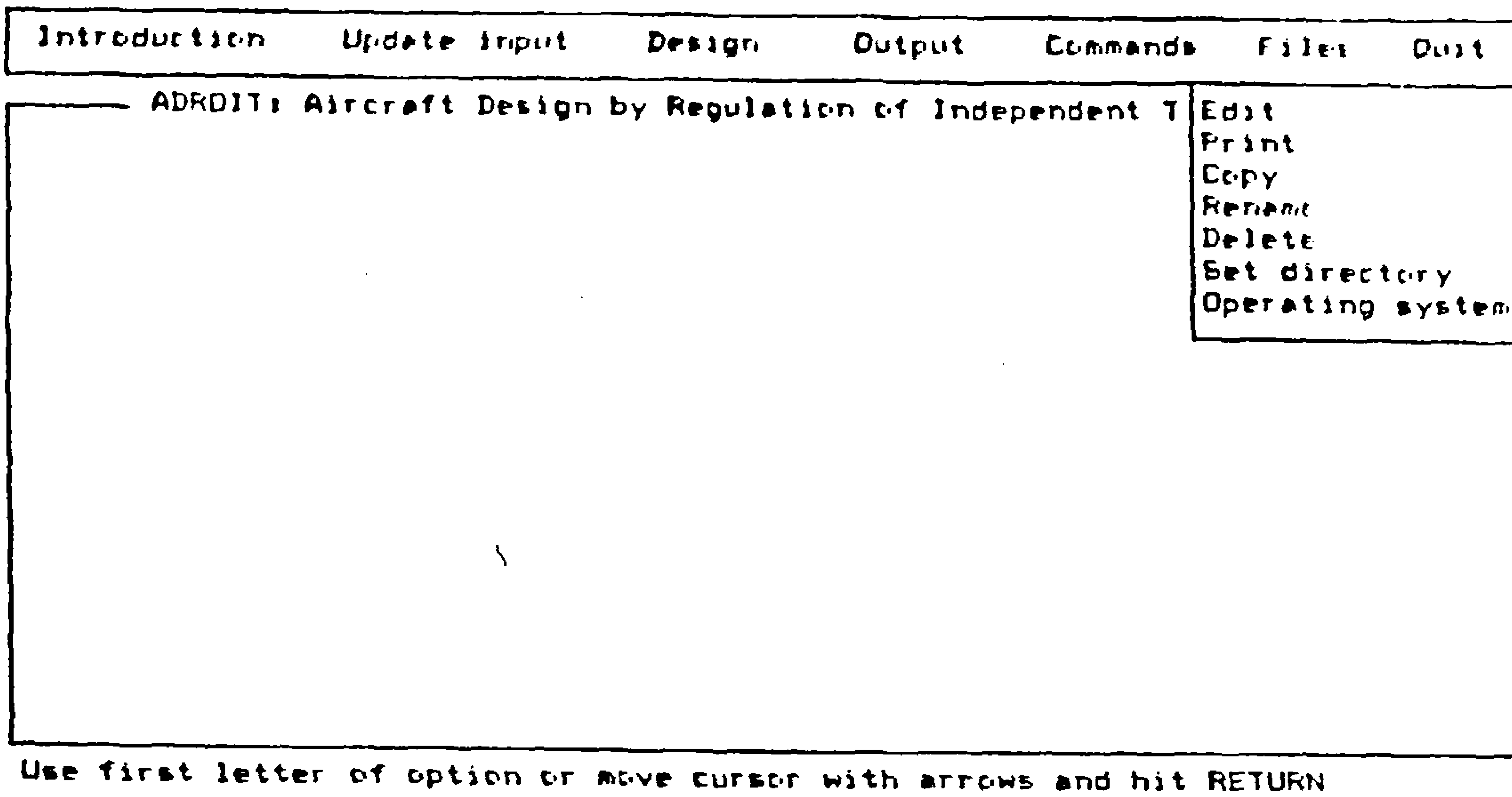
The Local spanwise CL distribution option displays the wing semi-span local lift coefficient together with the best section lift coefficient at stall as shown.



6.3.2.5 <u>Commands menu</u> - allows the user to perform various commands directed at controlling the design evaluation. A three option menu is offered as shown below.

```
Introduction   Update input   Design   Output   Commands   Files   Quit

ADROIT: Aircraft Design by Regulation of I   Undo previous design step(s)
                                             Restore a previous design
                                             Save present design

Use first letter of option or move cursor with arrows and hit RETURN
```

6.3.2.6 <u>Files menu</u> - supports various system (MS-DOS) specific commands as shown below.



6.3.2.7 Quit command - terminates the consultation session after confirming with the user.

6.3.3 Procedures for answering questions and checking input data

All input data follows a specific format easily interpreted from the system status line or help facility. The system checks that the answer to a specific question corresponds to a valid input format or condition. These questions are of the following type:

6.3.3.1 Menu input - In this area errors can occur when selecting an invalid option. The system displays the corresponding error message and repeats the menu.

Single option menu. The arrow keys can be used to indicate a choice and the F10 or Return key used to select it. The user can also select a given item from a menu by pressing the highlighted letter of that item.

Multiple option menu. The arrow keys can be used to indicate a choice from the menu and press Return to select (or de-select) it. F10 indicates that all the desired selections have been made. The user can also select a given item from a menu by pressing the highlighted letter of that item.

Pull-down menu. In this case the arrow keys move the cursor to an item in the main menu line and upon pressing Return, another menu appears, pulled down vertically below containing items closely related to the horizontal heading. Besides pressing Return, the user can also select a given item by pressing the highlighted letter of that item.

6.3.3.2 Line input - displays a window and allows input after the prompt. Once all the information is entered the user presses F10 or Return. The text can be edited using the arrow and delete keys. Line input errors can occur due to a typing error. The system displays the corresponding error message and repeats the question.

6.3.3.3 Screen input - is used to request input in a quick and uniform way by drawing a screen with a title and the questions text with its corresponding answers. The user moves the cursor across the screen with the arrow keys or by pressing Return. Within ADROIT there are three types of screen questions:

Numeric questions are answered by typing over a previously given answer and can be edited using the arrow and delete keys. Screen input errors can only occur when answering a numeric question if the answer given lies outside a valid range or contains an invalid character (e.g., a letter). If this condition occurs, the system 'beeps' as soon as the user tries to answer (i.e., move to) another question or press F10 to end.

Menu questions are answered in the manner described above, and the menu is activated by pressing Return over the answer.

Yes/no questions are answered by pressing Return over the answer which replaces the previous answer with its opposite.

6.3.4 Requesting help

At any point during a consultation session the user can find the relevant help by pressing the F1 key which displays text relevant to the particular question. Also, a status line is associated with every input form to remind the user of the key actions. For example, during a screen input a two option menu is offered as shown below to provide

```
                          WING DESIGN INPUT DATA
                          **********************
                               ┌── Help ──┐
Nominal Operating Range (nm)   │Meaning    │ptimum Cruise Altitude (ft)   38000
                               │Example(s) │
Maximum Cruise Mach Number     └───────────┘conomic Cruise Mach Number    0.75

Cruise Lift Coefficient         0.57    Landing Lift Coefficient          2.6

Wing Aspect Ratio               9.9     Wing Taper Ratio                  0.3

Wing Twist (deg)                3       Maximum Wing Loading (Kg/m2)      500

All Up Mass (Kg)                53000


Fuselage Wing Interaction       Underwing Pod Engines        Active Controls
BAe 125                         yes                          no

Use first letter of option or move cursor with arrows and hit RETURN
```

Selecting the Meaning option displays the definition of the question with its valid range as shown below.

```
                         WING DESIGN INPUT DATA
                         **********************

Nominal Operating Range (nm)  3000      Optimum Cruise Altitude (ft)  38000

Maximum Cruise Mach Number    0.78      Economic Cruise Mach Number   0.75

Cruise Lift Coefficient       0.57      Landing Lift Coefficient      2.6

------------ Meaning -------------  9.9      Wing Taper Ratio              0.3
Operating Range
The aircraft range at maximum       3        Maximum Wing Loading (Kg/m2)  500
passenger and baggage payload.
Valid range = 500..9000 nm          53000
************END**************

                                    Underwing Pod Engines     Active Controls
                                    yes                       no

F2:Goto line    F3:Search    S-F10:Resize window    F10:End
```

Selecting the Examples option displays typical high subsonic aircraft values as shown below.

```
                         WING DESIGN INPUT DATA
                         **********************

Nominal Operating Range (nm)  3000      Optimum Cruise Altitude (ft)  38000

Maximum Cruise Mach Number    0.78      Economic Cruise Mach Number   0.75

Cruise Lift Coefficient       0.57      Landing Lift Coefficient      2.6

------- Example(s) -------      9.9     Wing Taper Ratio              0.3
Operating Range (nm)
Aircraft            Value      3       Maximum Wing Loading (Kg/m2)  500
A310-200             3500
A320-200             2000      53000
B737-300             1615
B757-200             2310
BAe 125-800          2870      Underwing Pod Engines     Active Controls
CHALLENGER 600       3123      yes                       no
CITATION III         2540
DC10-30              4000

F2:Goto line    F3:Search    S-F10:Resize window    F10:End
```

6.4 CONSULTATION SESSION

This section traces a consultation session with ADROIT indicating the calculations being performed at the various design stages. It draws together the concepts introduced in this Chapter and clearly shows how the program can be employed complementing the earlier work. The input data used in this session corresponds to ADROIT's default answers shown in the previous section under screen input. In order to start a session the user types ADROIT and presses Return where upon the pull-down menu shown below is offered.

```
Introduction    Update input    Design    Output    Commands    Files    Quit

ADROIT: Aircraft Design by Regulation of Independent Tasks (ja)

Use first letter of option or move cursor with arrows and hit RETURN
```

The actual design session commences by selecting the Design option.

```
Introduction    Update input    Design    Output    Commands    Files    Quit
                                 Select design step
ADROIT: Aircraft Design b        exit               ent Tasks (ja)
                                 wing_design

Use first letter of option or move cursor with arrows and hit RETURN
```

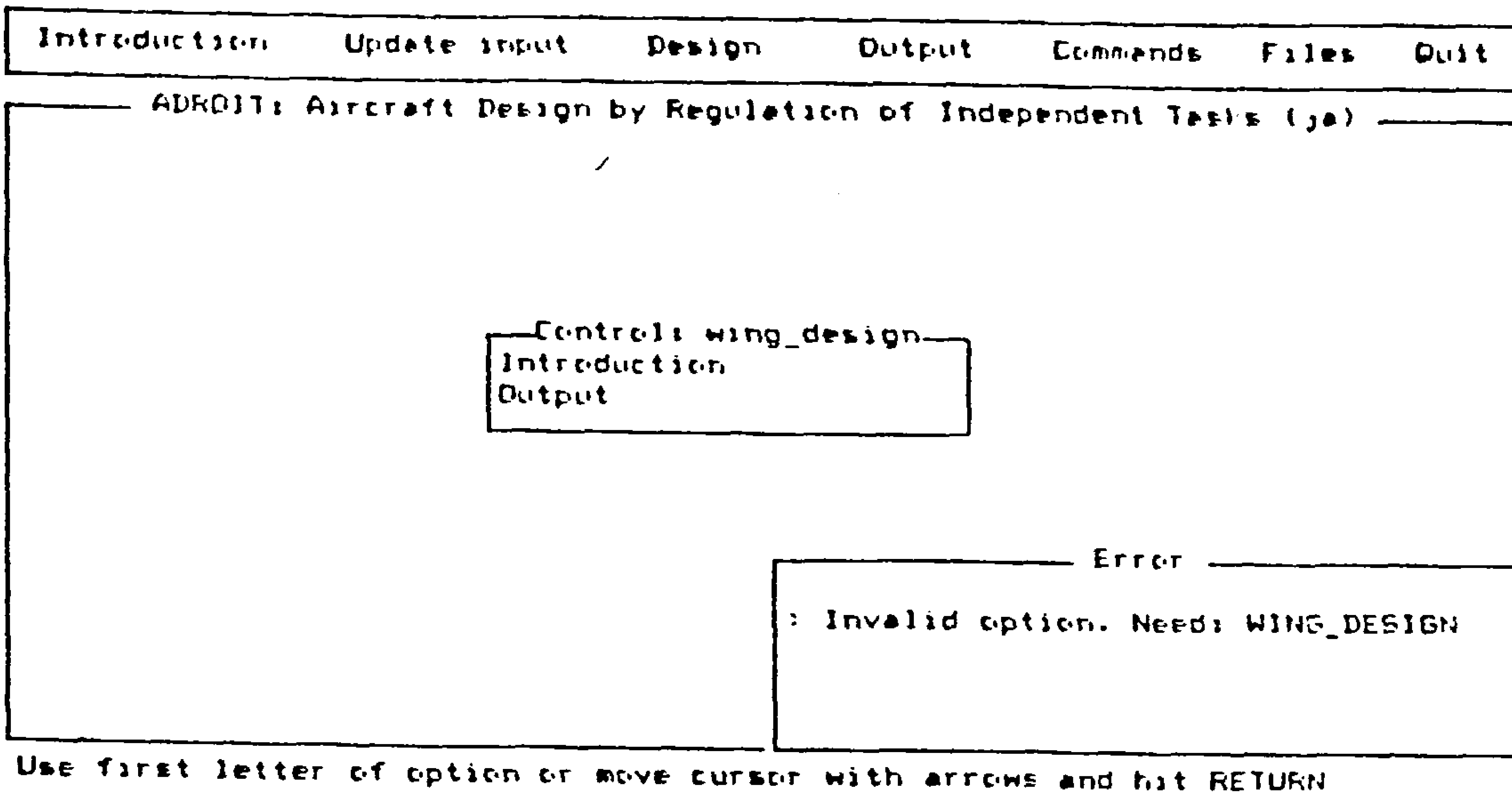
We may now select wing_design to move into the wing design process otherwise we may 'exit'. The system will now present the following menu:

```
Introduction    Update input    Design    Output    Commands    Files    Quit

——— ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ———

                ——Control: wing_design——
                Introduction
                Output

Use first letter of option or move cursor with arrows and hit RETURN
```

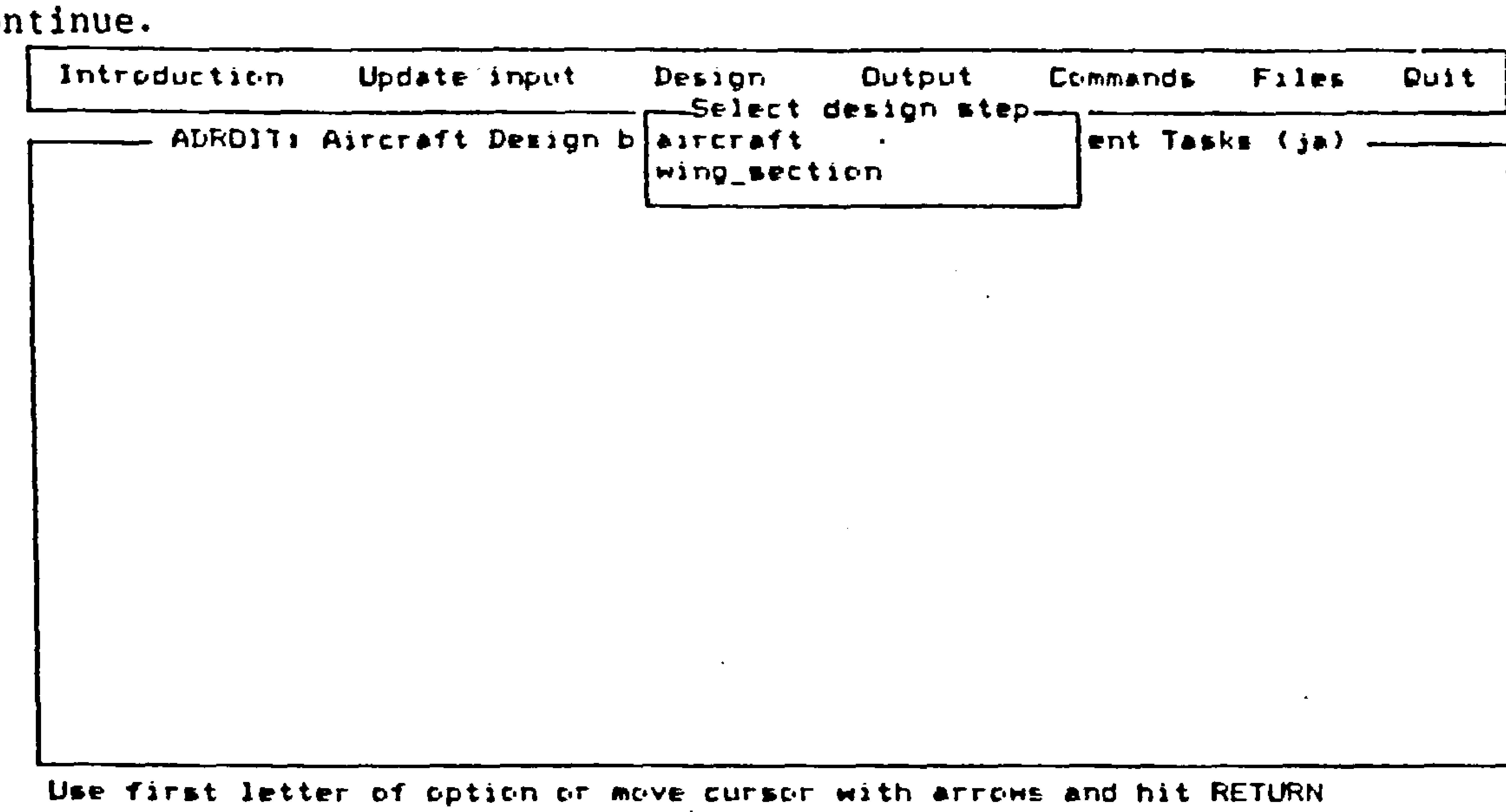
We shall now select Introduction from this control menu. (Note that during a consultation session any introduction to a control menu can be viewed).

```
Introduction    Update input    Design    Output    Commands    Files    Quit

——— ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ———

          ————————— Introduction —————————
                    WING DESIGN
                    ***********

          INTRODUCTION
          ------------
          Wing design is currently implemented in the
          ADROIT computer program which follows the
          design procedure used by the expert designer.

          SCOPE
          -----
          The scope of design is further restricted to
          the use of:

                 - Three supercritical aerofoil sections
                   suitable for high subsonic speeds are
                   used.

F2:Goto line    F3:Search    S-F10:Resize window    F10:End
```

The above introduction describes the scope of the wing design procedure and can be viewed at leisure using the arrow keys. We now image that we press F10 to end.



After returning to the control menu, if the Output option is selected an error occurs as this step has not yet been evaluated. Press Esc to continue.



ADROIT continues with the design by computing the next design steps to be evaluated within wing_design i.e., wing_section. We select the wing_section option. (Note that the aircraft option or any parent design step can be used to go up the design tree).

```
 Introduction    Update input    Design     Output     Commands    Files    Quit

 ------ ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ------

                        ----Action: wing_section----
                        Introduction
                        Update input and re-evaluate
                        Output

 ------------ Message ------------
 > Doing: wing_section

 Use first letter of option or move cursor with arrows and hit RETURN
```

The message 'Doing: wing_section' is displayed and the action menu for this .design step is offered. Selecting the Introduction option gives the following:

```
 Introduction    Update input    Design     Output     Commands    Files    Quit

 ------ ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ------

              -------------- Introduction --------------
                          WING SECTION CHOICE
                          *******************
              From the data available there is a choice to be
              made between three supercritical sections at the
              10.5% t/c ratio (RAE 9515 and 9530) and at 12.2%
              (RAE 9550) in terms of:

                         1.    Low Speed Lift Coefficient At Stall
                         2.    Stall Behaviour at Low Speed
                         3.    Lift Coeficient
                         4.    Pitching Moment
                         5.    2D Drag Rise Mach Number

 > Doing: wing_  NB: The relative importance of each of the vario
                 factors used in designing a wing need to be take
                 into account.  Thus, the term Relative Importanc
                 is used to indicate the importance of a specific

 F2:Goto line     F3:Search     S-F10:Resize window     F10:End
```

The above introduction describes the scope of the wing section procedure and can be viewed at leisure using the arrow keys. Pressing F10 ends this option:

```
Introduction    Update input    Design    Output    Commands    Files    Quit

ADROIT: Aircraft Design by Regulation of Independent Tasks (ja)

                    Action: wing_section
                    Introduction
                    Update input and re-evaluate
                    Output

Message
: Doing: wing_section

Use first letter of option or move cursor with arrows and hit RETURN
```

From the above menu we select the Update input and re-evaluate option.

```
WING SECTION CHOICE INPUT DATA
******************************

Economic Cruise Mach Number:        0.75

Cruise Lift Coefficient:            0.57

Fill in details. Move cursor with arrows. RETURN:Select F1:Hlp F10:End ESC:Quit
```

The above screen shows the input necessary to the wing_section template as described above, and which has been defined in the Update input option of the main menu.

Economic cruise Mach number (Mecon) = 0.75

Cruise lift coefficient (CL) = 0.57

We press Esc to return to the action menu.

```
Introduction    Update input    Design    Output    Commands    Files    Quit

------ ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ------

                    ----Action: wing_section----
                    Introduction
                    Update input and re-evaluate
                    Output

------ Message ------
: Doing: wing_section

Use first letter of option or move cursor with arrows and hit RETURN
```

We may now proceed by selecting Output to view the results file.

```
Introduction    Update input    Design    Output    Commands    Files    Quit

------ ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ------
                          ------ Output ------
                         WING SECTION CHOICE
                         *******************

                    STALL LIFT COEFFICIENT
                    **********************
        From the wing section data (at low speed),
        the values/scores obtained are:

        Section        Stall Lift Coeff.     Score
        RAE 9515            1.030             0.00
        RAE 9530            1.270            12.00
        RAE 9550            1.080             2.50

        I nominate the section RAE 9530 as the best section
        for this case study. The scores have been computed
        using a difference of 0.2 to represent a relative
        importance of 10 towards the evaluation of the best
        section.

F2:Goto line    F3:Search    S-F10:Resize window    F10:End
```

The above window shows the stall lift coefficient evaluation. For each section the stall lift coefficient at low speeds is obtained from experimental data and using equation 1 the ratings are evaluated. For example, in section RAE 9515

$$i = (10/0.2) * (1.03-1.03) = 0$$

We can press pgDn (page down) to continue viewing the results.

```
 Introduction    Update input    Design    Output    Commands    Files    Quit

 ------ ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ------
                            ----------- Output -----------
                    STALL BEHAVIOUR
                    ***************
                From the wing section data (at low speed),
                the values/scores obtained are:

                Section      Slope Lift Coeff.     Score
                RAE 9515          0.930              7.20
                RAE 9530          0.858              0.00
                RAE 9550          0.963             10.50

                I nominate the section RAE 9550 as the best section
                for this case study. The scores have been computed
                using a difference of 0.1 to represent a relative
                importance of 10 towards the evaluation of the best
 > Doing: wing_ section.

                    DRAG COEFFICIENT
                    ****************
 F2:Goto line    F3:Search    S-F10:Resize window    F10:End
```

The above window shows the evaluation of the stall lift coefficient behaviour. The slope at the stall is obtained for each section as described above and equation 2 is used to rate the sections. For example, in section RAE 9515

$$i = (10/0.1) * (0.930-0.858) = 7.2$$

We can continue viewing the results by pressing PgDn.

```
 Introduction    Update input    Design    Output    Commands    Files    Quit

 ------ ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ------
                            ----------- Output -----------
                    DRAG COEFFICIENT
                    ****************
                From the wing section data (at cruise conditions),
                the values/scores obtained are:

                Section      Drag Coefficient     Score
                RAE 9515          0.016             6.48
                RAE 9530          0.015             8.22
                RAE 9550          0.018             0.00

                I nominate the section RAE 9530 as the best section
                for this case study. The scores have been computed
                using a difference of 0.005 to represent a relative
                importance of 15 towards the evaluation of the best
 > Doing: wing_ section.
 > Evaluating:
 > Doing: wing_     PITCHING MOMENT
                    ***************
 F2:Goto line    F3:Search    S-F10:Resize window    F10:End
```

The above window shows the drag coefficient evaluation. For each section the drag coefficient at Mecon=0.75 and CL=0.57 is obtained by extrapolating or interpolating between the different experimental graphs and using equation 3 the section ratings are evaluated. For example, in section RAE 9515

$$i = (15/0.005) * (0.018-0.016) = 6.48$$

We press pgDn to continue viewing the results.

```
 Introduction    Update input    Design    Output    Commands    Files    Quit

 ——— ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ———

                        ——————————— Output ———————————
                               PITCHING MOMENT
                               ***************
                        From the wing section data (at cruise conditions),
                        the values/scores obtained are:

                        Section        Pitching Moment      Score
                        RAE 9515           -0.096            0.88
                        RAE 9530           -0.101            0.00
                        RAE 9550           -0.075            5.11

                        I nominate the section RAE 9550 as the best section
                        for this case study. The scores have been computed
                        using a difference of 0.05 to represent a relative
 > Doing: wing_         importance of 10 towards the evaluation of the best
 > Doing: wing_         section.

                                 2D DRAG RISE

 F2:Goto line     F3:Search     5-F10:Resize window     F10:End
```

The above window shows the pitching moment evaluation. For each section the pitching moment at Mecon=0.75 and CL=0.57 is obtained by extrapolating or interpolating between the different experimental graphs and using equation 4 the ratings are evaluated. For example, in section RAE 9515

$$i = (10/0.05) * (0.101-0.096) = 0.88$$

To continue viewing the results we employ pgDn.

```
 Introduction   Update input   Design   Output   Commands   Files   Quit

 ADROIT: Aircraft Design by Regulation of Independent Tasks (ja)

                                 Output
              2D DRAG RISE
              ************
 From the wing section data (at cruise conditions),
 the values/scores obtained are:

 Section        2D Drag Rise        Score
 RAE 9515          0.770            38.78
 RAE 9530          0.748            32.41
 RAE 9550          0.640             0.00

 I nominate the section RAE 9515 as the best section
 for this case study. The scores have been computed
 using a difference of 0.1 to represent a relative
 importance of 30 towards the evaluation of the best
 section.

> Doing: wing_
> Doing: wing_

F2:Goto line   F3:Search   S-F10:Resize window   F10:End
```

The above window appears and shows the 2D drag rise evaluation. For each section the 2D drag rise at CL=0.57 is obtained using a polynomial equation of the experimental graph and equation 4 is used to rate each section. For example, in section RAE 9515

$$i = (30/0.1) * (0.77-0.64) = 38.78$$

Once more we press pgDn to continue viewing the results.

```
 Introduction   Update input   Design   Output   Commands   Files   Quit

 ADROIT: Aircraft Design by Regulation of Independent Tasks (ja)

                                 Output
 using a difference of 0.1 to represent a relative
 importance of 30 towards the evaluation of the best
 section.

                    OVERALL RATINGS
                    ***************
          Section                       Total score
          RAE 9515                        53.35
          RAE 9530                        52.63
          RAE 9550                        18.11

 In accordance to your requirements I have choosen
 as the overall best section RAE 9515

 *******************END***********************

> Doing: wing_
> Doing: wing_

F2:Goto line   F3:Search   S-F10:Resize window   F10:End
```

The above window shows the overall section ratings found by summing the parameter ratings for each section. RAE 9515 is the best section as it has the highest rating. At this stage we press F10 to end this design task.

```
Introduction    Update input    Design    Output    Commands    Files    Quit

ADROIT: Aircraft Design by Regulation of Independent Tasks (ja)

                Action: wing_section
                Introduction
                Update input and re-evaluate
                Output

Use first letter of option or move cursor with arrows and hit RETURN
```

To continue with the design we now press Esc.

```
Introduction    Update input    Design    Output    Commands    Files    Quit
                                Select design step
ADROIT: Aircraft Design b       aircraft              ent Tasks (ja)
                                sweep_angle

            Message
> Doing: wing_section
> Evaluating: wing_section

Use first letter of option or move cursor with arrows and hit RETURN
```

ADROIT evaluates the wing_section choice by checking that over-extrapolation has not occurred and continues with the design by computing the next design steps to be evaluated within wing_design, i.e., sweep_angle. So we select the sweep_angle option:

```
 Introduction     Update input     Design     Output     Commands     Files     Quit

 ------ ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ------

                         --Control: sweep_angle--
                         Introduction
                         Output

 ---------- Message ----------
 > Doing: wing_section
 > Evaluating: wing_section

 Use first letter of option or move cursor with arrows and hit RETURN
```

Press Esc to continue.

```
 Introduction     Update input     Design     Output     Commands     Files     Quit
                                  --Select design step--
 ------ ADROIT: Aircraft Design b wing_design            ent Tasks (ja) ------
                                  drag_rise_3d

 ---------- Message ----------
 > Doing: wing_section
 > Evaluating: wing_section

 Use first letter of option or move cursor with arrows and hit RETURN
```

The sweep_angle control template has other sub-steps as shown in Figure 4.2, drag_rise_3d being the first. We continue by selecting the drag_rise_3d option:

```
 Introduction    Update input    Design    Output    Commands    Files    Quit

 ------- ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) -------

                    Action: drag_rise_3d
                    Introduction
                    Update input and re-evaluate
                    Output

 ---------- Message ----------
 > Doing: wing_section
 > Evaluating: wing_section
 > Doing: drag_rise_3d

 Use first letter of option or move cursor with arrows and hit RETURN
```

The message 'Doing: drag_rise_3d' is displayed and an action menu for this design step is offered. By selecting the Output option we obtain the following output displays:

```
 Introduction    Update input    Design    Output    Commands    Files    Quit

 ------- ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) -------
                         ------- Output -------
                              3D DRAG RISE
                              ************
            From your design conditions and the quoted fuselage
            wing blending the most likely combinations of sweep
            and t/c to satisfy 3D economic cruise Drag Rise are

            Sweep Angle      t/c (average)    Valid
             15              10.28            T
             20              10.42            T
             25              10.60            T
             30              10.83            T
             35              11.10            T
 ---------- M 40             11.41            T
             45              11.77            T
 > Doing: wing_s NB: The valid combination of sweep back angle and
 > Evaluating: w t/c is taken to be when the average t/c is less
 > Doing: drag_r than 18%.

 F2:Goto line     F3:Search     S-F10:Resize window     F10:End
```

```
 Introduction    Update input    Design    Output    Commands    Files    Quit

 ──── ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ────
                              ──── Output ────
                  40              11.41          T
                  45              11.77          T
                 NB: The valid combination of sweep back angle and
                 t/c is taken to be when the average t/c is less
                 than 18%.


                 The high speed requirement is satisfied when MDR
                 (economic cruise Mach number + decrement due to
                 the fuselage blending) is within 0.02 of Mmax
                 (maximum cruise mach number).
 ────────────── M
                 MDR        Mmax       High speed requirement
 > Doing: wing_s 0.80       0.78             .  T
 > Evaluating: w
 > Doing: drag_r ********************END************************

 F2:Goto line     F3:Search     S-F10:Resize window     F10:End
```

The above two windows show the 3-D drag rise evaluation as described in section 6.2. The user has defined:

wing-fuselage interaction = BAe 125

Maximum cruise Mach number = 0.78

The reduction (Decr) in 3-D drag rise Mach number is found from equation 6 i.e., Decr=0.05, and the sections are generated using equation 9:

$$t/c = (t/c)1 \frac{(MD - MD2)}{(MD1 - MD2)} + (t/c)2 \frac{(MD - MD1)}{(MD2 - MD1)}$$

Where

$(t/c)1 = 0.105$ (best section t/c RAE 9515)

$(t/c)2 = 0.122$ (2nd best section t/c RAE 9550 with <> t/c)

$MD = Mecon + Decr = 0.75 + 0.05 = 0.8$

$$MD1 = (MD1)_{\Lambda=0} / \cos^{1/2}\Lambda_{1/4}$$

$$MD2 = (MD2)_{\Lambda=0} / \cos^{1/2}\Lambda_{1/4}$$

$(MD1)_{\Lambda=0} = 0.77$ *(from wing section)*

$(MD2)_{\Lambda=0} = 0.64$ (from wing section)

$\Lambda 1/4$ = sweep angles (15,20,25,30,35,40,45)

For example, at $\Lambda 1/4 = 25$

$$t/c = 0.105 \frac{(0.8 - 0.748/\cos 25)^{1/2}}{\left(\frac{(0.77 - 0.748)}{\cos 25}\right)^{1/2}} + 0.122 \frac{(0.8 - 0.77/\cos 25)^{1/2}}{\left(\frac{(0.748 - 0.77)}{\cos 25}\right)^{1/2}}$$

$= 0.106$ (i.e., t/c average)

Test high speed requirement using:

$Mmax - MDR =< 0.02$

Where

$Mmax = 0.78$
$MDR = Mecon + Decr = 0.75 + 0.05 = 0.8$

At this point we press F10 to end this part of the design process.

The system responds with the following:

```
Introduction   Update input   Design   Output   Commands   Files   Quit

------ ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ------

                ----Action: drag_rise_3d----
                Introduction
                Update input and re-evaluate
                Output

Use first letter of option or move cursor with arrows and hit RETURN
```

We press Esc to continue with the design.

```
Introduction     Update input     Design     Output     Commands     Files     Quit
                                          Select design step
ADROIT: Aircraft Design b  wing_design                     ent Tasks (ja)
                           aeroelastic
                           flap

                Message
> Doing: wing_section
> Evaluating: wing_section
> Doing: drag_rise_3d
> Evaluating: drag_rise_3d
Use first letter of option or move cursor with arrows and hit RETURN
```

ADROIT evaluates drag_rise_3d by checking that the high speed requirement is satisfied, and that the sections generated have a t/c less than 18%. The design is continued by computing the next design steps to be evaluated within sweep_angle i.e., aeroelastic and flap. We select the aeroelastic option.

```
Introduction     Update input     Design     Output     Commands     Files     Quit
ADROIT: Aircraft Design by Regulation of Independent Tasks (ja)

                      Action: aeroelastic
                      Introduction
                      Update input and re-evaluate
                      Output

                Message
> Doing: wing_section
> Evaluating: wing_section
> Doing: drag_rise_3d
> Evaluating: drag_rise_3d
> Doing: aeroelastic
Use first letter of option or move cursor with arrows and hit RETURN
```

The message 'Doing: aeroelastic' is displayed and an action menu for this design step is offered. By selecting the Output option we obtain:

```
Introduction   Update input   Design   Output   Commands   Files   Quit

ADROIT: Aircraft Design by Regulation of Independent Tasks (ja)
                              Output
            AEROELASTIC STIFFNESS
            *******************
The torsion and bending stiffness are satisfied
with the following aspect ratios:

Angle   t/c     Torsion   Bending   Combined   Valid
 15    10.28     9.90      9.90      9.90        T
 20    10.42     9.90      9.89      9.89        T
 25    10.60     9.90      9.76      9.76        T
 30    10.83     9.90      9.60      9.60        T
 35    11.10     9.90      9.41      9.41        T
 40    11.41     9.90      9.17      9.17        T
 45    11.77     9.90      8.87      8.87        F

NB: The combined results for the Torsion and Bending
stiffness are satisfied by taking a default tolerance
on the initial aspect ratio variance equal to 10%.
Initial aspect ratio =   9.90

F2:Goto line     F3:Search     S-F10:Resize window     F10:End
```

The above window shows the results found during the evaluation of the aeroelastic check. The user has defined the following parameters:

engine position = underwing mounted
target aspect ratio (A) = 9.9
cruise altitude (H) = 38000 nm
active controls = no

The torsion stiffness check using wing mounted engines is:

$$\frac{A^{3/2}}{t/c^{2}} < \frac{3 \times 10^{8}}{VD^{2} \cos \Lambda 1/4}$$

The bending stiffness check with no active controls is:

$$\frac{A^{3/2} \sec \Lambda 1/4}{t/c} < \frac{850}{3.75}$$

For $\Lambda 1/4$ = 25 the above values and inequalities are:

A = 9.9 (from user)

H = 38000 ft (user input)
T = 216.7 K for H > 36000 ft

$$Vsound = 1.9438 * (401.8 * T)^{1/2} = 573.57 \text{ Knots}$$

MD = Mmax + 0.05 = 0.78 + 0.05 = 0.83

$VD = MD * Vsound = 0.83 * 573.57 = 476.06$ Knots

$(t/c)a = 0.106$ (from 3-D drag rise)

$t/c = (t/c)r = (t/c)a * 1.4 = 0.14863$

The torsion stiffness check is:

$$\frac{9.9^{3/2}}{(0.14863)^2} < \frac{3 \times 10^8}{476.06^2 \cos 25}$$

$$1422.1 < 1460.57$$

The bending stiffness check is:

$$\frac{9.9^{3/2} \sec 25}{0.14863} < \frac{850}{3.75}$$

$$232.28 < 226.67$$

The latter inequality is not satisfied, use A = 9.76.

$$\frac{9.76^{3/2} \sec 25}{0.14863} < \frac{850}{3.75}$$

$$226.61 < 226.67$$

Note that the sweep angle at 45 degrees has been eliminated since the corresponding aspect ratio has been reduced by more than 10% in order to satisfy the inequalities. To end the output option we press F10 and return to the main part of the program:

```
Introduction    Update input    Design    Output    Commands    Files    Quit

------ ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ------

                   Action: aeroelastic
                   Introduction
                   Update input and re-evaluate
                   Output

Use first letter of option or move cursor with arrows and hit RETURN
```

We press Esc to continue with the next design stage.

```
Introduction    Update input    Design    Output    Commands    Files    Quit
                                  Select design step
------ ADROIT: Aircraft Design b wing_design         ent Tasks (ja) ------
                                 flap
                                 tip_stall
                                 wing_weight

              Message
> Evaluating: wing_section
> Doing: drag_rise_3d
> Evaluating: drag_rise_3d
> Doing: aeroelastic
> Evaluating: aeroelastic

Use first letter of option or move cursor with arrows and hit RETURN
```

ADROIT evaluates the aeroelastic design step by checking that there are valid sweep angles i.e., those which yield aspect ratios within 10% of the target aspect ratio. The design is continued by computing the next design steps to be evaluated within sweep_angle i.e., flap, tip_stall, and wing_weight. Select the tip_stall option.

```
Introduction    Update input    Design    Output    Commands    Files    Quit

ADROIT: Aircraft Design by Regulation of Independent Tasks (ja)

                    Action: tip_stall
                    Introduction
                    Update input and re-evaluate
                    Output

Message
> Doing: drag_rise_3d
> Evaluating: drag_rise_3d
> Doing: aeroelastic
> Evaluating: aeroelastic
> Doing: tip_stall

Use first letter of option or move cursor with arrows and hit RETURN
```

The message 'Doing: tip_stall' is displayed and an action menu for this design step is offered. Select the Output option.

```
Introduction    Update input    Design    Output    Commands    Files    Quit

ADROIT: Aircraft Design by Regulation of Independent Tasks (ja)
                                Output
                    TIP STALL
                    *********
Notation:
          CLy         Local Lift Coefficient
          N (2y/b)    non-dimensional semi-spanwise position
          y           semi-spanwise position
          b           wing span
          A           quarter chord sweep angle
```

N \ A	15	20	25	30	35	40	45
0	0.49	0.47	0.46	0.44	0.43	0.41	
0.1	0.51	0.50	0.49	0.47	0.46	0.45	
0.2	0.53	0.52	0.51	0.50	0.49	0.48	

```
F2:Goto line    F3:Search    S-F10:Resize window    F10:End
```

Introduction Update input Design Output Commands Files Quit

ADROIT: Aircraft Design by Regulation of Independent Tasks (ja)

Output

0.3	0.56	0.55	0.54	0.54	0.53	0.52	
0.4	0.58	0.57	0.57	0.57	0.57	0.56	
0.5	0.60	0.60	0.60	0.60	0.61	0.61	
0.6	0.61	0.62	0.63	0.64	0.64	0.65	
0.7	0.61	0.63	0.64	0.66	0.68	0.70	
0.8	0.58	0.60	0.63	0.65	0.67	0.70	
0.9	0.47	0.50	0.52	0.54	0.57	0.60	
1	0.00	0.00	0.00	0.00	0.00	0.00	
Valid	T	T	T	T	T	T	F

F2:Goto line F3:Search S-F10:Resize window F10:End

Introduction Update input Design Output Commands Files Quit

ADROIT: Aircraft Design by Regulation of Independent Tasks (ja)

Output

0.7	0.61	0.63	0.64	0.66	0.68	0.70	
0.8	0.58	0.60	0.63	0.65	0.67	0.70	
0.9	0.47	0.50	0.52	0.54	0.57	0.60	
1	0.00	0.00	0.00	0.00	0.00	0.00	
Valid	T	T	T	T	T	T	F

NB: A valid angle is taken when the local lift coefficient (CLy) does not exceed the cruise stall Lift Coefficient (CLs), found by evaluating the stall lift coefficient at the 2-D drag rise.
CLs = 0.88

*********************END***********************

F2:Goto line F3:Search S-F10:Resize window F10:End

The above three windows show the evaluation of the tip stall check. The user defined values are:

wing twist = 3
taper ratio = 0.3

Equations 14, 15a, 15b, and 16 are used to evaluate the local lift coefficient distribution CL(y) along the wing semispan. For example, at Λ1/4 = 25 and η = 0.7 the local lift coefficient is CL(y) = 0.64 using the following values:

a_0 = 6 1/rad

A = 9.76 at /\1/4 = 25 (from aeroelastic)

$$c(y) / \bar{c} = 3\,(0.7\ 0.3^2 - 0.7 + 1 + 0.3) / 2\,(0.3^2 + 0.3 + 1) = 0.715$$

$\overline{CL}$ = 0.57 (user input)

$m = 1 - M^2 = 1 - 0.75^2 = 0.4375$

M = Mecon = 0.75 (user input)

$\bar{y} = 0.42 + A\,m\,[(4.4 + 5\ )\ \tan/\backslash 1/4/m + 10.4^{1/2} - 6.7]/10^3$

α_o = -0.18 degrees (section RAE 9515 experimental data)

K = 0.5 (correction factor)

$\epsilon = - \eta * \epsilon_o = -0.7 * 3 = -2.1$ degrees

λ = 0.3 (user input)

The test performed consists in evaluating the best section cruise stall lift coefficient CLs = 0.88 from experimental data and checking that the local lift coefficient along the span for each sweep angle is less than this value.

$$CL(y) < CLs$$

$$0.64 < 0.88 \quad \text{for } \Lambda 1/4 = 25 \text{ and } \eta = 0.7$$

To end this feature we press F10

```
 Introduction    Update input    Design    Output    Commands    Files    Quit

 ------ ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ------

                      ------Action: tip_stall------
                      Introduction
                      Update input and re-evaluate
                      Output

 Use first letter of option or move cursor with arrows and hit RETURN
```

and Esc to continue with the design.

```
 Introduction    Update input    Design     Output    Commands   Files   Quit
                                  ┌Select design step┐
 ─── ADROIT: Aircraft Design b │wing_design         │ent Tasks (ja) ───
                                  │flap                │
                                  │wing_weight         │
                                  └────────────────────┘

 ──────────── Message ────────────
 > Evaluating: drag_rise_3d
 > Doing: aeroelastic
 > Evaluating: aeroelastic
 > Doing: tip_stall
 > Evaluating: tip_stall
 Use first letter of option or move cursor with arrows and hit RETURN
```

ADROIT evaluates tip_stall by checking that there are valid sweep angles, and the design is continued by computing the next design steps to be evaluated within sweep_angle i.e., flap and wing_weight. We continue by selecting the flap option.

```
 Introduction    Update input    Design     Output    Commands   Files   Quit
 ─── ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ───

                        ┌──────Action: flap──────┐
                        │Introduction              │
                        │Update input and re-evaluate│
                        │Output                    │
                        └──────────────────────────┘

 ──────────── Message ────────────
 > Doing: aeroelastic
 > Evaluating: aeroelastic
 > Doing: tip_stall
 > Evaluating: tip_stall
 > Doing: flap
 Use first letter of option or move cursor with arrows and hit RETURN
```

The message 'Doing: flap' is displayed and an action menu for this design step is offered. Select the Output option.

```
Introduction   Update input   Design   Output   Commands   Files   Quit

 ADROIT: Aircraft Design by Regulation of Independent Tasks (ja)
                         Output
   ADROIT
   File: FLAP.OUT                          Date: 12: 4:1988
   Disk: C:\                               Time: 11:42:26

            FLAP EFFECTIVENESS
            ******************
   Sweep    Max. Lift      Valid
   Angle    Coefficient

     15         2.90          T
     20         2.82          T
     25         2.72          T
     30         2.60          F
     35         2.46          F
     40         2.30          F
     45         2.12          F

   ********************END************************

F2:Goto line    F3:Search    S-F10:Resize window    F10:End
```

The above window shows the results found during the evaluation of the flap effectiveness check. The user defines the maximum landing lift coefficient (CLmax = 2.6) and the following relation is used to check the flap effectiveness:

CLmax =< 3 cosΛ1/4

For example when Λ1/4 = 25 then

2.6 =< 3 cos25

2.6 =< 2.72

To end we press F10 as usual.

```
Introduction   Update input   Design   Output   Commands   Files   Quit

 ADROIT: Aircraft Design by Regulation of Independent Tasks (ja)


                   Action: flap
                   Introduction
                   Update input and re-evaluate
                   Output


Use first letter of option or move cursor with arrows and hit RETURN
```

Again we press Esc to continue with the design.

```
Introduction     Update input     Design     Output     Commands     Files     Quit
                                 Select design step
------- ADROIT: Aircraft Design b wing_design          ent Tasks (ja) -------
                                  wing_weight

------------- Message -------------
> Evaluating: aeroelastic
> Doing: tip_stall
> Evaluating: tip_stall
> Doing: flap
> Evaluating: flap
Use first letter of option or move cursor with arrows and hit RETURN
```

ADROIT evaluates flap by checking that there are valid sweep angles, and the design is continued by computing the next design steps to be evaluated within sweep_angle i.e., wing_weight. We now select the wing_weight option.

```
Introduction     Update input     Design     Output     Commands     Files     Quit
------- ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) -------

                        Action: wing_weight
                        Introduction
                        Update input and re-evaluate
                        Output

------------- Message -------------
> Doing: tip_stall
> Evaluating: tip_stall
> Doing: flap
> Evaluating: flap
> Doing: wing_weight
Use first letter of option or move cursor with arrows and hit RETURN
```

The message 'Doing: wing_weight' is displayed and an action menu for this design step is offered. Select the Output option.

```
 Introduction    Update input    Design     Output     Commands    Files    Quit

 ------ ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ------
                               Output
                 ADROIT
                 File: WING WEIGHT.OUT             Date: 12: 4:1988
                 Disk: C:\                         Time: 11:46:36

                                 WING WEIGHT
                                 ***********
                 Sweep      t/c      Aspect       Wing
                 Angle    average    Ratio        Weight (kg)

                  15       10.28      9.90          4936.64
                  20       10.42      9.89          5026.24
                  25       10.60      9.76          5123.14
             M    30       10.83      9.60          5248.25
 > Doing: tip_st  35       11.10      9.41          5406.52
 > Evaluating: t  40       11.41      9.17          5604.90
 > Doing: flap
 > Evaluating: f  *******************END*********************
 > Doing: wing_w

 F2:Goto line     F3:Search     S-F10:Resize window     F10:End
```

The above window shows the evaluation of the wing weight. The user has defined:

Maximum wing loading (W/S) = 500 Kg/m2
All up mass (W) = 53000 Kg
Operating range (R) = 3000 nm

The relation used to evaluate the wing weight is:

$$W_w = C\left[\frac{bS}{\cos\Lambda 1/4}\left(\frac{1+2\lambda}{3+3\lambda}\right)\left(\frac{WN}{S}\right)^{0.3}\left(\frac{VD}{t/c}\right)^{0.5}\right]^{0.9}$$

For example, at $\Lambda 1/4 = 25$ the wing weight (Ww) is 5123.14 Kg using

A	=	9.76 (from aeroelastic)
C	=	C1 (R - R1)/(R1 - R2) + C2 (R - R2)/(R2 - R1) = 0.0322
C1	=	0.034
C2	=	0.028
N	=	3.75 (no active controls)
S	=	W/(W/S) = 53000/500 = 106 m2
R	=	3000 nm (user input)
R1	=	500 nm
R2	=	9000 nm
t/c	=	(t/c)r = 0.11484 (from 3-D drag rise)
VD	=	244.91 m/s (from aeroelastic)
W	=	53000 Kgs (user input)
W/S	=	500 Kgs/m2 (user input)
λ	=	0.3 (user input)
$\Lambda 1/4$	=	25 degrees (example calculation)
b S	=	$S^{3/2} A^{1/2} = (106)^{3/2} (9.76)^{1/2} = 3409.45\ m^3$

We end by pressing F10.

```
Introduction    Update input    Design    Output    Commands    Files    Quit

ADROIT: Aircraft Design by Regulation of Independent Tasks (ja)

                 Action: wing_weight
                 Introduction
                 Update input and re-evaluate
                 Output

Use first letter of option or move cursor with arrows and hit RETURN
```

Press Esc to continue with the design.

```
Introduction    Update input    Design    Output    Commands    Files    Quit

ADROIT: Aircraft Design b  Are the results satisfactory (y/n) ? y

Message
> Evaluating: tip_stall
> Doing: flap
> Evaluating: flap
> Doing: wing_weight
> Evaluating: wing_weight

Use first letter of option or move cursor with arrows and hit RETURN
```

As there are no built in checks for wing_weight, ADROIT confirms with the user whether or not the results are satisfactory. We shall regard them as satisfactory hence type y and press Return.

```
Introduction     Update input     Design     Output     Commands     Files     Quit

 ──── ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ────

                        ┌Control: sweep_angle──┐
                        │Introduction          │
                        │Output                │
                        └──────────────────────┘

 ───────── Message ──────────
> Evaluating: tip_stall
> Doing: flap
> Evaluating: flap
> Doing: wing_weight
> Evaluating: wing_weight
Use first letter of option or move cursor with arrows and hit RETURN
```

ADROIT computes the next design steps to be evaluated within sweep_angle and displays the control template for sweep_angle. Following our normal practice we select the Output option.

```
Introduction     Update input     Design     Output     Commands     Files     Quit

 ──── ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ────
                   ───────────── Output ─────────────

                             SWEEP ANGLE
                             ***********
                  After performing the various checks the status o
                  the various sweep back angles are:

                  Sweep    Aeroelastic    Tip      Flap
                  Angle    Stiffness      Stall    Effectiveness
                    15         T            T        T
                    20         T            T        T
                    25         T            T        T
                    30         T            T        F
 ────────────── M   35         T            T        F
> Evaluating: t     40         T            T        F
> Doing: flap       45         F            F        F
> Evaluating: f
> Doing: wing_w   ******************END**********************
> Evaluating: w
F2:Goto line     F3:Search     S-F10:Resize window     F10:End
```

The above window shows the valid and invalid sweep angles for the various checks performed by the program. Press F10 to end and obtain:

```
Introduction    Update input    Design    Output    Commands    Files    Quit

——— ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ———

                    Control: sweep_angle
                    Introduction
                    Output

Use first letter of option or move cursor with arrows and hit RETURN
```

Press Esc to continue.

```
Introduction    Update input    Design    Output    Commands    Files    Quit

——— ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ———

                    Control: wing_design
                    Introduction
                    Output

Use first letter of option or move cursor with arrows and hit RETURN
```

ADROIT computes the next design steps to be evaluated and offers the control template for wing_design. We now select out final Output option.

```
 Introduction    Update input    Design    Output    Commands    Files    Quit

 ---------- ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ----------
                    ----------------- Output -----------------
                    ADROIT
                    File: WING DESIGN.OUT            Date: 12: 4:1988
                    Disk: C:\                        Time: 11:53:25

                           WING DESIGN
                           ***********
                    According to your design conditions the followin
                    results have been obtained:

                      1.  Best 2D aerofoil section is RAE 9515

                      2.  Valid sweep back angle(s):    15    20    25
                    *******************END************************

 F2:Goto line     F3:Search     S-F10:Resize window     F10:End
```

The above window shows the best wing section and the valid range of sweep angles. We press F10 to end.

```
 Introduction    Update input    Design    Output    Commands    Files    Quit

 ---------- ADROIT: Aircraft Design by Regulation of Independent Tasks (ja) ----------

                          --Control: aircraft--
                          Introduction
                          Output

 Use first letter of option or move cursor with arrows and hit RETURN
```

ADROIT computes the next design steps to be evaluated and offers the control template for aircraft. The above Output option will display all the results calculated during the session. To exit from the design process we press 'exit'.

At this stage, ADROIT displays the message 'Finished: aircraft' and returns to the main menu. The user can use the various options to re-view results, undo design steps, print result files etc.. When finished select the Quit command from the main menu to return to MS-DOS.

6.5 CONCLUSIONS

Throughout the development of the ADROIT program for wing design, two considerations have been dominant:

1. To create a practical application in which the template concept for knowledge representation can be explored and identify the limitations inherent with this approach.

2. To create a practical program which can be easily used by both expert and novice designer.

In its present form the program can perform the wing design of a subsonic airliner both at high and low subsonic speeds. The user can control the extend of the consultation and, explore rapidly and effectively different wing designs.

The following Chapter considers the limitations and capabilities of the present implementation within the scope of the objectives set out in this research thesis.

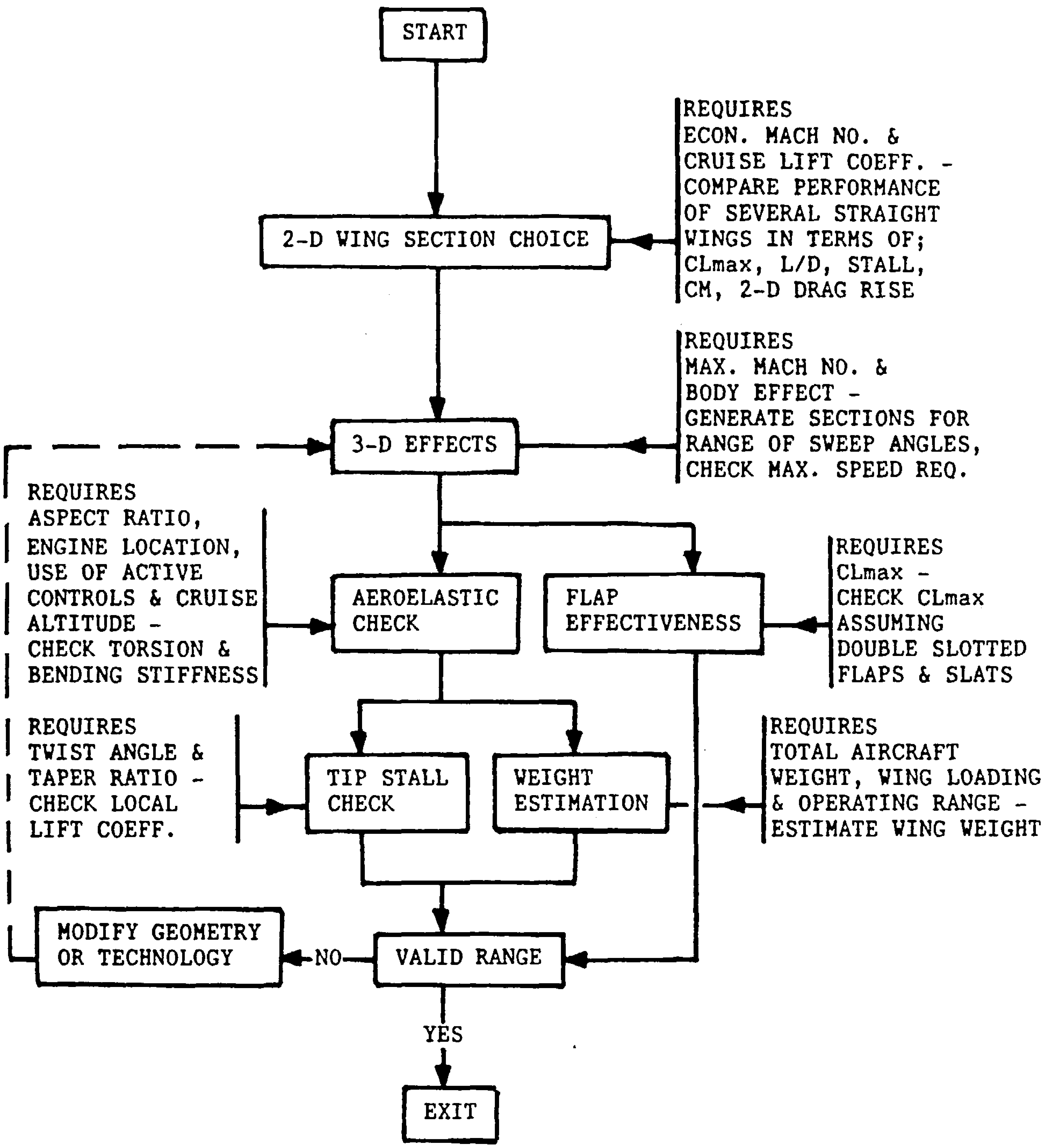


FIGURE 6.1 STAGES IN THE WING DESIGN PROGRAM

	PARAMETER	RELATIVE DIFFERENCE	RELATIVE IMPORTANCE	
$C_{l\,max}$	LIFT COEFFICIENT	0.2	10	
$C_{l\alpha}$	STALL BEHAVIOUR	0.5	~~8~~	= 15
C_D	DRAG COEFFICIENT	0.005	15	
C_m	PITCHING MOMENT	0.05	10	
M_{DD}	2-D DRAG RISE	0.1	30	

FIGURE 6.2. TABULAR WING SECTION RATINGS

CHAPTER 7

LIMITATIONS OF TRIAL IMPLEMENTATION

7.1 INTRODUCTION

Chapter 3 described the funtionality of the various stages in wing design. During Chapter 4 this initial study was complemented by identifying the procedures and judgements used by the designer. Based on this study a framework for an aircraft design expert system was proposed in Chapter 4 and the template concept was defined to represent the design knowledge as small and independent design steps. Wing design was implemented using this framework whose scope, procedures, and operating instructions have been described in Chapter 6. This implementation has been verified against worked design examples (1) and has been used by design students and lecturers at Cranfield to test and generate their respective wing designs. The aim of this Chapter is to draw from this implementation the limitations of the approach and present the lessons towards the implementation of further aircraft design modules.

7.2 LIMITATIONS

The current wing design program has a number of problematic features which future systems should concentrate on improving.

1. ADROIT stores the results of calculations and the choices of components in a global database as described in Chapter 4. For example, the computed aerodynamic data in the wing_section step are stored as the three argument PROLOG fact data(SYMBOL,SECTION,VALUE). These values are accessed by several steps and their retrieval can be slow. A great improvement would be achieved if the symbol name was used.

2. There is no explanation to the user of the reasoning structure employed in the design i.e., commands to display the reason for a question, how a conclusion has been made, and why a different solution path was not taken. This is because the rules and algorithms are too procedural and rely heavily on side effects for their operation. Any explanation facility would need to rely heavily on canned text in order to describe a particular design step.

3. The backtracking mechanism provided makes the whole system very slow. The essential problem is that PROLOG is tied to chronological backtracking. A different approach to revision of deductions is needed for example, a reasoning maintenance system to record for each choice all previously made choices on which it depends and the inconsistent set of choices.

4. The fixed numerical importance associated with the wing section parameters does not always represent the correct weightings since they are dependent on various factors such as aircraft type, cruise speed etc.

5. The use of only three supercritical sections introduces some innaccuracies in the design process as described in Chapter 6. A greater number of sections including standard sections would improve this situation. However, access to supercritical section aerodynamic data is difficult due to commercial confidiality.

7.3 LESSONS FOR AIRCRAFT CONFIGURATION

Chapter 5 addressed the knowledge representation aspects concerning aircraft configuration and enumerated the conceptual differences with wing design. Based on this analysis, the extensions necessary to the aircraft design expert system framework described in Chapter 4 were identified in order to model the aircraft configuration problem. The problems with the current program provide further refinements to these extensions.

1. Wing design is a well understood activity consisting of mainly procedural knowledge. A declarative formalism for representing this knowledge would allow the design process to be followed and described during a tutorial i.e., declarative representation of the formulae and heuristics used in wing design is necessary if good explanation is to be achieved.

2. A forward looking capability is required as a design decision is most naturally explained in terms of its eventual impact on parts of the design that have not been completed.

3. Knowledge within the system should be used to reduce the selection of the next step and similarly, the selection of the backtrack step when the design fails should be reduced to one. Only when several choices remain after reduction should the selection be left to the user. In the latter case, the capability to 'learn' new reduction techniques from the user answers should be provided. This learning capability has been identified in Chapter 2 as an integral part of the design process.

4. The current implementation stores the graphical data within the PROLOG database, this data should be stored in an external database and manipulated by PROLOG via a procedural language in order to improve the efficiency.

5. An Object-Oriented approach to handle the representation of graphical and algebraic data in a similar fashion is required.

6. A more versatile system for making numerical judgements is required to take into account requirements and different aircraft types.

7.4 CONCLUSIONS

The difficulties with the current implementation as described above may impede further development of the expert system as other modules are incorporated. For modules such as aircraft configuration the template concept becomes unsuitable as a knowledge representation mechanism because of the interrelation between the different steps and the fact that a single path through these steps cannot be formulated in advanced. Based on these limitations recommendations have been put forward towards further development of the expert system in order to incorporate aircraft configuration. The following Chapter presents the overall conclusions of the research work carried out and lists a number of extensions to the system for future work.

CHAPTER 8

CONCLUDING DISCUSSION AND RECOMMENDATIONS

8.1 INTRODUCTION

The outlines of an expert system for aircraft design have been described. This has been made possible by a detailed analysis of the aircraft design knowledge base following a two pass approach. The first pass consisted in analysing the wing design knowledge base and creating a prototype program using a framework for an aircraft design expert system. During the second pass, a detailed analysis of the aircraft configuration knowledge base was performed which delivered a specification for the modifications and extensions necessary to the framework developed during the wing design implementation. The approach achieves the objectives of the current research in having effectively defined the aircraft design knowledge base and providing tools and specification for new tools which are to be used within the aircraft design expert system.

8.2 OBJECTIVES ACHIEVED

The work has been directed at providing an expert system which will assist engineers in the preliminary design of a civil airliner. In order to achieve worthwhile results, the scope of the design tasks has been considerably limited.

The status of the work is one where an actual working program for wing design exists with supporting documentation and a very effective examination and evaluation of the knowledge base performed based on the investigation of the aircraft design process, particularly wing design and aircraft configuration. The latter steps represent important parts of aircraft design and show the rich nature of the aircraft design knowledge. This provides a firm base from which further work can be carried out.

8.3 RECOMMENDATIONS FOR FUTURE WORK

Some further development areas which can be pursued as part of the continuing research effort towards the development of an aircraft design expert system are now discussed.

Further design steps need to be implemented i.e., all aspects of aircraft configuration and other modules as described in Chapter 3. This development needs to be carefully integrated with the wing design and the fuselage design (76) programs. In this way the enhanced program will be able to perform the full range of aircraft design tasks.

Object-Oriented programming represents a powerful method of encapsulating knowledge about objects and how they interact in the world as outlined in Chapter 1. This approach should be investigated with view at supporting the development of effective representation and reasoning tools for the aircraft design expert system.

Application oriented programs. Attention should be given to the incorporation of applications oriented programs into the system with particular emphasis given to:

1. The incorporation of the Finite Element (FE) method into the design process will require the definition and evaluation of the solution process. Based on modelling rules used by the FE analyst, the selection of a set of finite elements suitable for the problem, definition of a satisfactory grid of elements (mesh) which can geometrically define the structure in space and finally, the principles used in the interpretation of the computed results.

2. In order to extend the scope of the design program an evaluation of the current Computer Aided Design (CAD) systems is required to assess suitable programs to link with the expert system.

Man-machine intereface as described in Chapter 5 represents one of the most challenging areas in the developement of an aircraft design expert system thus, more effective methods for interacting with the user are required employing an engineering oriented interface. For example, an Object-Oriented approach for representing aircraft parts as objects lends itself to a better integration when assembling the overall aircraft.

Tutorial and explanation facilities will require defining a graded series of 'help' like facilities which relate to the level of expertise of the user and an 'assessor' program to monitor the users use of the system and increase or decrease the help provided.

Extensive explanation facilities should be provided to allow users to interrogate the systems inference and reasoning process. This will require a graphical interface to provide appropriate outputs.

Other design domains. Finally, the techniques and tools developed for an aircraft design expert system should be generalised with view to cover a broader design base to include non-aircraft regimes such as cars, building, ships etc.

8.4 CONCLUSIONS

The work carried out in the present research has been directed at providing an expert system which will assist engineers in the preliminary design of civil airliners. Existing computer applications in this area take control of the task and the designer has essentially no control over the computational process once the design problem is formulated. Futhermore, the designer has no way of obtaining explanation for the solutions reached.

The present research was motivated by the desire to apply AI methodologies to the design process. First, because the design task itself is becoming increasingly complex with the use of new materials, methods of production and concepts within an increasingly competitive environment. Secondly, the rapid increase in the use of specialist applications software such as FE packages requires a high level of experience if errors are to be avoided.

A modest effort has been made to cover these aspects which will help further the development of a suitable design expert system. Some directions for further work have also been indicated.

REFERENCES

1. HOWE, D., and FIELDING, J.P., Lecture notes on aircraft design, Cranfield Institute of Technology, England, 1984.

2. TORENBEEK, E., Synthesis of Subsonic Airplane Design, Delft University Press,

3. CORNING, G., Supersonic and Subsonic, CTOL and VTOL, Airplane Design, 1976.

4. STINTON D., The anatomy of the aeroplane, Collins, 1983.

5. NICOLAI, L.M., Fundamentals of aircraft design, METS, Inc., 1975.

6. KUCHENMANN, D., The aerodynamic design of aircraft, Pergamon, 1978.

7. LUNDSTROM, S., and LARSEN, R.L., The potential state of computer and information technology in the year 2000, Hawaii International Conference on System Sciences, Jan. 1985.

8. BARR, A., FEIGENBAUM, E.A., and COHEN, P.R. (editors), Handbook of Artificial Intelligence, 3 vols., William Kaufmann, 1982.

9. WINSTON, P.H., Artificial Intelligence, 2nd Edition, Addison-Wesley, 1984.

10. CHARNIAK, E., and McDERMOTT, D., Introduction to Artificial Intelligence, Addison-Wesley, 1985.

11. SIMONS, G.L., Introducing Artificial Intelligence, NCC Publications, 1984.

12. HAYES, J.E., and MICHIE, D. (editors), Machine Intelligence 10 - Intelligent Systems: Practice and Perspective, Ellis Horwood, 1982.

13. HAYES-ROTH, F., WATERMAN, D.A., and LENAT, D., Building Expert Systems, Addison-Wesley, 1983.

14. WEISS, S.M., and KULIKOWSKI, C.A., A Practical Guide to Designing Expert Systems, Chapman and Hall, 1983.

15. SELL, P.S., Expert Systems - A Practical Introduction, MacMillan, 1985.

16. FORTHSYTH, R. (ed.), Expert Systems: Principles and Case Studies, Chapman and Hall, 1984.

17. MORRIS, A.J., and FIELDING, J.P., LOKI report: Stage by stage layout of preliminary wing design for cruise conditions, Cranfield Institute of Technology, Bedford, England, 1983.

18. HADDAWY, P., and FREKSA, C., LOKI Report KR2: A description of knowledge representation features for a design support system, Technical University of Munich, Munich, West Germany, May 1984.

19. SAGE User Manual V.1.3, Knowledge Engineering Group, Systems Programming Limited, 1983.

20. SAGE Language Manual V.1.3, Knowledge Engineering Group, Systems Programming Limited, 1983.

21. ALSINA, J., Evaluation of SAGE: An Expert System Shell, Cranfield Institute of Technology, Bedford, England, 1984.

22. CLOCKSIN, W.F., and MELLISH, C.S., Programming in Prolog, Springer-Verlag, 1981.

23. STERLING, L., and SHAPIRO, E., The Art of Prolog: Advanced Programming Techniques, MIT Press, 1986.

24. BRATKO, I., Programming for Artificial Intelligence, Addison-Wesley, 1986.

25. ALSINA, J., and BRUYNOOGHE M., LOKI report: A simple/general controller and an explanation facility for the ADROIT program, Cranfield Institute of Technology, Cranfield, England, and BIM, Everberg, Belgium, 1985.

26. BATES, P., LOKI AD3 report: ADROIT's Programmer Manual, Scicon Ltd., London, England, 1985.

27. ALSINA, J., LOKI AD3 report: ADROIT's User Manual, Cranfield Institute of Technology, Bedford, England, 1985.

28. MORRIS, A.J., and BUCHANAN, A.G., ADROIT: Aircraft Design by Regulation of Independent Tasks, Proc. of IKBS in Defence Seminar, Royal Signals Research Establishment, Malven, England, 1984.

29. ALSINA, J., FIELDING, J.P., and MORRIS, A.J., ADROIT: An Aircraft Design Aid, ESPRIT'85: A Status Report on Continuing Work, North-Holland, 1985.

30. ALSINA, J., FIELDING, J.P., and MORRIS, A.J., An expert system for aircraft design, Presented at the 3rd Int. Symposioum on Optical and Optoelectronic Applied Sciences and Engineering, Innsbruck, Austria, 1986.

31. ALSINA, J., FIELDING, J.P., and MORRIS, A.J., Progress towards an aircraft design expert system, in Computer Applications in Aircraft Design and Operation, eds. T.K.S. Murthy and J.P. Fielding, Springer-Verlag, 1987.

32. ALSINA, J., ANINOS, A., EROL, N., FIELDING, J.P., and MORRIS, A.J., Progress towards a commercial aircraft design expert system, presented at DGLR conference, Berlin, W. Germany, October 1987.

33. NILSSON, N.J., Principles of Artificial Intelligence, Springer-Verlag, 1982.

34. NEWELL, A., and SIMON, H.A., GPS, a program that simulates human thought, in Computers and Though, Feigenbaum, E.A., and Feldman, J. (eds.), McGraw-Hill, 1963.

35. SAMUEL, A.L., Some studies in Machine Learning using the game of checkers, in Computers and Though, Feigenbaum, E.A., and Feldman, J. (eds.), McGraw-Hill, 1963.

36. MICHALSKI, R.S., CARBONELL, J.G., and MITCHELL, T.M. (eds.), Machine Learning: an Artificial Intelligence Approach, Tioga Publishing Company, 1983.

37. DUDA, R.O., and SHORTLIFFE, E.H., Expert Systems Research, Science, vol. 220, 1983, pp 261-267.

38. FOX, J. (ed.), Expert Systems 83, BSC Special Interest Group on Expert Systems, 1983.

39. WATERMAN, D., A Guide to Expert Systems, Addison-Wesley, 1985.

40. SHEIL, B., Power Tools for Programmers, Datamation, vol. 29, Feb. 1983, pp 131-144.

41. McDERMOTT, J., R1: A Rule-Based Configurer for Computer Systems, Artificial Intelligence, vol. 19, no. 1, 1982.

42. HART, P.E., DUDA, R.O., and EINAUDI, M.T., PROSPECTOR: A Computer-Based Consultation System for Mineral Exploration, Mathematical Geology, vol. 10, no. 5, 1978.

43. SHORTLIFFE, E.H., Computer-Based Medical Consultations: MYCIN, Elsevier, 1976.

44. LINDSAY, R., BUCHANAN, B.G., FEIGENBAUM, E.A., and LEDERBERG, J., Applications of Artificial Intelligence for Chemical Inference: The DENDRAL Project, McGraw-Hill, 1980.

45. GRINBERG, M.R., A Knowledge-Based Design System for Digital Electronics, Proceedings of the 1st Annual National Conference on Artificial Intelligence, American Association for Artificial Intelligence, 1980.

46. MAHER, M. L., and FENVES, S. J., HIRISE: An Expert System for the Preliminary Structural Design of High Rise Buildings, in Knowledge Engineering in Computer Aided Design, Gero, J. (ed.), North-Holland, 1985.

47. BROWN, D.C., and CHANDRASEKARAN, B., Expert System for a Class of Mechanical Design Activity, Proceedings of the International Federation for Information Processing, WG5.2, Working Conference on Knowledge Engineering in Computer Aided Design, Hungary, 1984.

48. SWIFT, K., and MATTHEWS, A., Expert Computer Systems in Engineering Design, Engineering, September 1983.

49. BASKIN, C., (ed.), Theme: Representing knowledge, BYTE, Nov. 1986, pp 147-231.

50. BIBEL, W., BRUYNOOGHE, M., FREKSA, C., HADDAWY, P., and MOREAU, M., LOKI Report KR1: An overview of formalisms, issues, languages, and systems for the representation of knowledge, Technical University of Munich, Munich, West Germany, and BIM, Everberg, Belgium, Dec. 1983.

51. ROBINSON, J.A., A machine oriented logic based on the resolution principle, J. ACM, vol. 12, 1965, pp 23-41.

52. KOWALSKI, R., Predicate logic as a programming language, Proc. IFIP Congress, Stockholm, 1984.

53. KOWALSKI, R., Logic for Problem Solving, North-Holland, 1979.

54. MOSKOWITZ, L., Rule-Based programming, BYTE, Nov. 1986, pp.217-224.

55. GOLDBERG, A., Introducing the Smalltalk-80 system, BYTE, Aug. 1981.

56. GOLDBERG, A., and ROBSON, D., Smalltalk-80: The Language and its Implementation, Addison-Wesley, 1983.

57. PHILLIPS, C., Smalltalk V, Personal Computer World, Nov. 1986, pp 168-171.

58. WHITE, E., and MALLOY, R., (eds.), Theme: Object-Oriented Languages, BYTE, Aug. 1986, pp 136-233.

59. DIXON, J.R., Design Engineering: Inventiveness, Analysis, and Decision Making, McGraw-Hill, 1980.

60. WINOGRAD, T., Understanding Natural Language, Academic Press, 1972.

61. AEA, Association of European Airlines, Airliner Specification, May 1985.

62. EDWARDS, G.R., Problems in the development of a new aeroplane, Journal of the Royal Aeronautical Society, Vol. 53, 1949, pp 197-252.

63. FIELDING, J.P., And now - the plastic airliner. An advanced short range airliner design project, Cranfield Institute of Technology, Bedford, England, 1985.

64. ROLLS-ROYCE, The Jet Engine, Rolls-Royce LTD., 1966.

65. McDONNEL DOUGLAS, Propfan program - background information, News from McDonnel Douglas, May 1985.

66. MIDDLENTON, P., Frugal, fast, and fractions, Flight International, Aug. 84, pp 113-122.

67. LASER, How many engines and where?, Flight International, Vol. 94, Aug. 68, pp 206-207.

68. MAHONEY, C.F., Balance of Power, Flight International, Vol. 91, July 67, pp 1046-1047. .

69. HENDERSON, W.P. and ABEYOUNIS, W.K., Aerodynamic characteristics of a high-wing transport configuration with an over the wing nacelle-pylon arrangement, NASA TP 2497, Dec. 1985.

70. CAMERON-JOHNSON, A., The undercarriage in aeroplane project design, Aircraft Engineering, Vol. 41, Feb. 1969, pp 6-11.

71. CONWAY, H.G., Landing Gear Design, Clapman and Hall, 1958.

72. CURRY, N.S., Landing Gear Design Handbook, Lockheed-Georgia Company, 1982.

73. TAYLOR, R.T., and RAY, E.J., Factors affecting the stability of T-tail transports, Journal Aircraft, Vol. 3, 4, July-Aug. 1966.

74. ZIENKIEWICZ, O.C., The Finite Element Method, 3rd Edition, McGraw-Hill, 1977.

75. FULKER, J.L., Aerodynamic Data for Three Supercritical Aerofoils: RAE (NPL) 9515 and 9530, and RAE 9550, RAE 3820, Bedford, England, 1978.

76. CIPOLLONE, G.A.F.G., Development of an expert system for fuselage design, M.Sc. thesis, Cranfield Institute of Technology, Bedford, England, 1986.