CRANFIELD INSTITUTE OF TECHNOLOGY

COLLEGE OF AERONAUTICS

AEROSPACE VEHICLE DESIGN
(*ARTIFICIAL INTELLIGENCE GROUP*)

Ph. D. THESIS

ACADEMIC YEAR : OCT. 1, 1988 -- APRIL. 30, 1991

DESAID (THE DEVELOPMENT OF AN EXPERT SYSTEM FOR AIRCRAFT INITIAL DESIGN)

SEUNG-HYEOG NAH

SUPERVISOR : Professor A. J. Morris

JULY 25, 1991

(This thesis is submitted for the degree of Ph.D.)

DEDICATION

To My Ancestors in Heaven and
My Family in Korea

ACKNOWLEDGEMENT

I would like to express my sincere and deep appreciation to Professor A. J. Morris for his guidance, supervision, advice, and encouragement from the start of my research to the completion of this thesis.

Many thanks must be given to Professor D. Howe, Dr. J. P. Fielding and other staff at the College of Aeronautics for their invaluable help and kindness shown during my Cranfield life. Also all the help from Miss H. Hinds, Alex Mac Kinnon, and Mr. S. H. Macci is much appreciated.

I would especially like to thank my Managing Director H. S. Cho who decided to send me here for our company's future (even under its adversity), Dr. S.J. Kim, and Section manager J. E. Woo who took care of my numerous requests without complaint.

Finally, special thanks to my mother and family who always prayed for my health and successful accomplishment of my research during my stay at Cranfield, a gorgeously academic but at times lonesome campus.

SYNOPSIS

As all engineering works are a blend of theory and empiricism, aircraft design, by its nature, represents a mixture of aircraft designer's knowledge obtained from aeronautical engineering disciplines and its usage combined with his experience. This means not only the application but also the integration of all the fundamental knowledge of aerodynamics, structure, propulsion, stability and control, operational and economic aspects, etc., based upon the designer's judgements and experiences.

Thus the tasks involved in designing an aircraft configuration, without exception, show complex characteristics, considering the fact that aircraft configuration design means the integration of components such as lifting surfaces (wing), fuselage, power-plant, control surfaces (tail or canard), and undercarriage.

The discrepancies and mismatches among the aircraft components make the configuration design iterative, repetitive, and thus time - consuming. Such complexities of configuration design processes often require compromise, through trial and error, to resolve conflicts between the major design areas.

Moreover, it takes tens of years to become a experienced design expert whose sound judgement, based upon experience and profound knowledge, influences greatly the aircraft configuration design. The differences in judgements depend upon the designers' imagination and experience, and they are the cause of variations in aircraft configurations.

Therefore, the efforts were made to overcome those difficulties which hinder the aircraft designer from making the task of configuration design more efficient, and further to assist the aircraft designer in getting an easy and interactive preliminary aircraft configuration without always relying upon design experts. Hence the current research project is directed at the development of an expert system for aircraft design. This involves the use of Artificial Intelligence and its programming language called PROLOG (PROgramming in LOGic).

The research started from a thorough analysis of the major component design areas and has constructed an EXPERT SYSTEM to find out the efficient Control Mechanism

which can search intensively for the solutions to design problems for all types of aircraft; civil and military, subsonic and supersonic, conventional and unconventional, etc. In addition, users can have access to the explanations of important items such as a design process, terminology, equations, and results. The explanation facility is one of the most important functions of Expert Systems.

Partly due to the limit of computer capacity and partly due to the magnitude of laborious program execution at this stage, the system implementation has focused on the high - subsonic, conventional and jet transport aircraft categories. The approach taken was to find an efficient and effective control mechanism (i.e. an Inference Engine), which integrated the PARAMETRIC STUDY, WING DESIGN, FUSELAGE DESIGN, ENGINE DESIGN, TAIL DESIGN, UNDERCARRIAGE DESIGN, WEIGHT ANALYSIS AND COST ANALYSIS into a whole configuration system.

The comparison between Expert System results and existing aircraft such as Boeing 747, Airbus 300 series, BAe 146 series, McDonnell Douglas MD series, etc., showed the permissible ranges of error to be within about 10 %. Such results enable the Expert System to claim that it can act as a useful design tool for the aircraft designer in the initial stage of aircraft configuration design.

Finally, the author believes that the control mechanism devised for this Expert System can be used as a sound basis for extending the Expert System to include other types of aircraft and further to encompass spacecraft design, as the designer wishes.

LIST OF CONTENTS

CHAPTER 1 INTRODUCTION

CHAPTER 2 AIRCRAFT DESIGN

CHAPTER 3 LESSONS FROM PAST COMPUTERIZED SYSTEM DEVELOPMENTS FOR AIRCRAFT DESIGN

CHAPTER 4 ANALYSIS OF AIRCRAFT DESIGN KNOWLEDGE

CHAPTER 5 PROBLEM STRUCTURING AND STRATEGY FOR SOLUTION

FIGURES

NOMENCLATURE

SYMBOL	MEANING
A.I.	Artificial Intelligence.
A.R.	Aspect Ratio.
A_f	Fuselage Cross Section Area, ft^2.
A_t	Total Wetted Area, ft^2.
B	Breguet Factor.
BCAR	British Civil Airworthiness Requirement
C_D	Total Drag Coefficient.
$C_{D,i}$	Induced Drag Coefficient.
$\triangle C_{D,f}$	Increment in Profile Drag Coefficient due to Trailing Edge Flap Deflection.
$\triangle C_{D,g}$	Increment in Profile Drag Coefficient due to Landing Gear Extension.
$\triangle C_{D,s}$	Increment in Profile Drag Coefficient due to Slat Deflection.
$C_{D,0}$	Zero Lift Drag Coefficient in Clean Condition.
$C_{H,T}$	Horizontal Tail Volume Coefficient.
C_L	Lift Coefficient.
$C_{L,K}$	Lift Coefficient at Cruising Speed of Assumed Lift to Drag Ratio.
$C_{L,M}$	Lift Coefficient at Cruising Speed of Maximum Lift to Drag Ratio.

$C_{L,T}$	Take-Off Lift Coefficient.
$C_{L,A}$	Approach Lift Coefficient.
$C_{L,2}$	Second Climb Lift Coefficient.
C_r	Wing Root Chord, ft.
C_t	Wing Tip Chord, ft.
$C_{V,T}$	Vertical Tail Volume Coefficient.
Decr	Decreased Magnitude
Dia.	Fuselage Diameter, ft.
D.O.C.	Direct Operating Cost.
FAR	Federal Aviation Regulations.
L/D	Lift to Drag Ratio.
L_T	F.A.R. Take-Off Field Length, ft.
M.A.C.	Mean Aerodynamic Chord of Wing, ft.
M_{cr}	Cruising Mach Number.
M_D	3-Dimensional Drag Rise Mach Number.
R	Cruise Range, nautical mile(n.mi.)
Rho	Density, lb/ft^3.
S	Wing Area, ft^2.
$S_{H,T}$ or S_H	Horizontal Stabilizer Area, ft^2.
$S_{V,T}$ or S_V	Vertical Stabilizer Area, ft^2.
Tau	Thickness and Chord Ratio at wing root.
T_C	Max. Engine Thrust at Cruising, lbf.
T_O	Maximum Engine Thrust, lbf.
T.R.	Taper Ratio.
$\bar{U}$ or U_{bar}	Useful Load Fraction.
V_{cr}	Cruising Speed, knot.
V_D	Design Diving Speed, knot.
W_{fuel}	Aircraft Fuel Weight, lb.

$W_{F,comp}$	Fuselage Weight with Composite, lb.
W_{Fuse}	Aircraft Fuselage Weight, lb.
W_G	Aircraft Take-Off Gross Weight, lb.
W_H	Horizontal Stabilizer Weight, lb.
W_L	Aircraft Max. Weight at Landing, lb.
W_p	Payload, lb.
W_V	Vertical Stabilizer Weight, lb.
Z_H	Vertical Distance from Vertical Tail Root to Horizontal Tail, ft.

a	Speed of Sound, knot.
c	Specific Fuel Consumption, lb/lb/hr.
d	Fuselage Diameter, ft.
e	Oswald Efficiency Factor.
l	Fuselage Length, ft.
l/d	Fuselage Fineness Ratio.
$l_{H,T}$	Distance from M.A.C.$_{1/4}$ of Horizontal Stabilizer to M.A.C.$_{1/4}$ of Wing, ft.
$l_{V,T}$	Distance from M.A.C. $_{1/4}$ of Vertical Stabilizer to M.A.C. $_{1/4}$ of Wing, ft.
t/c	Airfoil Thickness to Chord Ratio.
t_r	Wing Root Chord Thickness, ft.
t_t	Wing Tip Chord Thickness, ft.
/\ 1/2	Sweep angle at a half chord, deg.
/\ 1/4	sweep angle at a quarter chord, deg.

CHAPTER 1

INTRODUCTION

> "*In the two years of configuration development, over two million man hours were expended to investigate various configurations and approaches to determine the optimum design. ....* " -- W. M. Magruder, Development of requirement, configuration and design for the Lockheed L-1011, SAE paper No. 680688. - [3].

CHAPTER 1 INTRODUCTION

1.1 Introduction

As is the case with other engineering design, aircraft configuration design(*) is the process of producing the description of a manufacturable aircraft which satisfies a set of requirements. Thus the design activity involves representation, manipulation, and understanding of aircraft configuration (i.e., *aircraft and its components*), relations among processes, standards and aeronautical engineering disciplines combined with creative imagination.

The designer's task in the aircraft configuration design is to apply all the fundamentals of aerodynamics, structures, propulsion units, stability and control, and economic aspects, based upon his experience and judgement.

Therefore, at the commencement of aircraft configuration design, designers must firstly have the knowledge about aircraft configuration design. Secondly, they must comprehend how the design knowledge is to be applied in order to describe successful configuration which meets our specific needs satisfactorily.

Regretfully, it is troublesome in aircraft configuration design for designers to obtain such a wide knowledge and to apply their expertise to configuration design. Experience gained in aircraft configuration design showed the complex characteristics of cyclic processes, iteration and repetitive trade-off's, therefore making the design very time-consuming. Moreover, it requires many years for aircraft designers to obtain expertise and sufficient experience to use it efficiently at will and with ease. Accordingly, it is generally known that a new aircraft design can take a decade or more from the

---

(*) This refers to the general layout of relevant aircraft components, the integrated external shape, its synthesized dimensions, and other relevant characteristics. It is intended to indicate either 'conceptual and preliminary' or 'initial' design.

initial design through to manufacture and full operation within a certain degree of safety.[1] [54]

In the mean time, aircraft configurations have undergone continuous changes due to either technological advances or fashions which create design problems. Problems also occur, when either the formulated configuration design disciplines or the established data are not available for tackling the design problems. In this case, there is no choice but to entirely rely upon "rule of thumb" judgements. Such differences in designer's judgements are the cause of sophisticated variations in the aircraft configuration, even for aircraft with the same intended roles (for example, *Boeing Stratojet and Avro Vulcan B-1*).

In addition to the different judgements, there are technological advances which make configuration design sophisticated. The appearance of advanced propulsion units (i.e., *high by-pass turbofan engine and propfan engine*) and the sweep concepts (i.e., *forward sweep or backward sweep wing*) have deepened the diversity of aircraft configuration. These increased the difficulties in choosing the best solution from the various configurations which were considered from every angle and respect. [3] [7]

Hence, it has long been a goal of an aircraft designer to deepen and broaden the complex design knowledge and to acquire experience in using this knowledge effectively for successful configuration design. As a mainstay of achieving the goal, computers have been widely used in the initial design stage for numerical analyses in such fields as aerodynamics, structural analysis, stability and control, weight prediction, cost estimation , etc. As a matter of fact, aircraft companies have developed computerised systems for preliminary design purpose such as SYNAC (SYNthesis of AirCraft) by General Dynamics [28], CPDS (Computerised Preliminary Design System) by Boeing [29], Application of Numerical Optimizations [30,31], GASP (General Aviation Synthesis Program) [32], etc.

Examining the computerized systems developed to date, it is true to say that the use of computers can greatly reduce both the complexity and time consumption in configuration design by assisting the designer in numerical analyses in the major design areas.

However, traditional computing techniques, directed at either numerical analysis or data processing, have their limitations in assisting the designer in configuration design, where he must make the best use of his knowledge and experience for reasoning to a solution when a compromise of conflicting requirements is required.

Fortunately, a new approach to computing, directed at simulating the process of a human reasoning, appeared as a result of efforts made since the 1970s' and such computers are being used increasingly in most fields of engineering design, under the name of **Artificial Intelligence** (abbreviated as *A. I.*) or **Expert System.** In aircraft design, there have been developed many Expert Systems such as ADROIT, PASS, DSIDES, etc. [1] [36] [38] [51] [52] [54] [56]

The experiences gained in developing such Expert Systems show that it is possible for a design expert to examine a variety of ideas about Aircraft Configuration in the initial stage and that he can escape from the traditional approach of "*2 team or 2 configuration design*". The "*2 team approach*" means that just two types from the many configuration candidates are considered for further development because of a long design lead time. [6]

In conclusion, the appearance of "Artificial Intelligence" offers two interesting possibilities. Firstly it can reduce the complexity of design processes by assisting the designer in finding a solution when conflict resolution is needed. Secondly, A. I. techniques made it possible for the expertise in specific domains to be transferred if they are logically structured and stored in the Artificial Intelligence Computer. [51, 57, 61, 79, 80, 82]

Accordingly, it is the purpose of this research to develop a trial Expert System for aircraft configuration design in the initial design stage. For verification, the experimental results were compared with existing aircraft.

1.2 Description of the Work

For the case of initial configuration design, the author examines, *in Chapter 2*, the diversity of aircraft configurations. It is found that there are many existing configurations and configuration types which are "under investigation". Thus, generating an Expert System in this complex environment requires answering questions related to : "What kind of design knowledge is needed ?", "How can the design knowledge be classified and classes connected with each other ?", "How can the designer use the knowledge for solving configuration design problems ?", "What type of solution strategy is required ?", "How can Artificial Intelligence Techniques help the strategy ?", and "What is A.I. ?".

In Chapter 3, current computerized systems for aircraft design are reviewed to get some practical knowledge for programming this expert system.

In Chapter 4, the various aircraft design disciplines are analysized to classify the knowledge associated with the design processes and the connections amongst these classes. Thus "Component Design Considerations", "Input and Output Data", and the "required processes" are described.

In Chapter 5, the author explains the complexity of configuration design problems when searching for a desired solution. This complexity is associated with the relationships among basic concepts for aircraft configuration, its components for integration, and design activities for sizing them. Thus configuration design problems are then structured in a manner which maps onto a solution and this structure is described. Once the structure is fully defined, a solution strategy(Inference Engine) is developed and described.

In Chapter 6, the structures of this expert system and its program are explained with the lessons from Chapter 3 incorporated. The trial implementation is also explained with the screens shown in Appendix V.

In Chapter 7, the developed expert system is tested by comparing its results with the configuration data of real existing aircraft. Then some discussion is added to provide directions for further improvements.

In Appendix I, the detailed analysis of aircraft design knowledge and process is described. The 'Rule Expressions' of the configuration design knowledge base are attached in *Appendix II* and their equivalent 'Prolog Expressions' are added in *Appendix III*. In *Appendix IV*, Control Mechanism is converted into Prolog Expression. The source program of aircraft design analysis which is converted into Turbo Prolog is attached in *Appendix V* for later use by another researchers and/or users. The trial implementation with the screens is shown in *Appendix VI*.

CHAPTER 2

AIRCRAFT DESIGN

" *An Expert is a person who, because of training and experience, is able to do things the rest of us cannot ---. Experts know a great many things and have tricks and caveats for applying what they know to problems and tasks; they are good at plowing through irrelevant information in order to get at the basic issues, and they are good at recognizing problems --.* " - E. Johnson, What kind of expert should a system be ?, The Journal of medicine and Philosophy, Vol. 8, pp 77 - 97, 1983. [61]

CHAPTER 2 AIRCRAFT DESIGN

2.1 Diversity of Aircraft Configurations

In aircraft configuration design, both the development of the Jet engine (i.e., *Turbojet, Turbofan, and Propfan*) and the introduction of *Swept-back / Delta* wings have greatly widened the choice of aircraft configuration. Some examples include the de Havilland Comet, Boeing B-707 and its derivatives, Douglas DC-8 and its descendants, Swedish SAAB Viggin, Convair 7002/XF-92A, Douglas F4D-1 Skylay, Concorde, and Lockheed SR-71. In addition, there exist occasionally revolutionary or unconventional concepts in such aircraft as the Curtis Wright CW-24, Northrop N-1/9 and XB-35 (similar to that of today's B-2 Bomber), Sikorsky V-300 which revolutionised the helicopter business, and Lockheed P-38. [7] [45]

Some of the configurations have been developed through *evolutionary changes* from the previous ones while others have been developed by adapting *revolutionary concepts* which radically deviate from the expected trends among existing aircraft.

A review of the civil transport aircraft to date shows that the appearance of Jet Propulsion units and the development of Swept-Back wing brought revolutionary changes in aircraft configuration design. These two factors made it possible to diversify the aircraft configurations. Even odd numbers of engines (i.e., 3 engines) came into existence. For example, Propeller-type transports had only either a single engine at the frontal part of the fuselage or wing mounted engines (i.e. DC-3/4, Lockheed Constellation and L-1011 Hercules). However, Jet-propelled Transport had both Under-Wing Mounted (i.e., Boeing B-707 and its series) and Rear-Fuselage-Mounted Engines (i.e., DC-9 and MD-80). Three engined aircraft with one engine mounted at the center of the rear fuselage (for example, B-727, Trident, DC-10 and its derivatives) also emerged.

As fuel cost was expected to increase, a new type of engine called the 'prop-fan' or UDF (Unducted Fan) began to be incorporated into Transport Configuration. The design of the MD-80 adopted the use of a prop-fan.

2.1.1 Review of Conventional Aircraft

Aircraft manufacturers are conservative and very reluctant to adopt either new technology or configurations which are very different from previous ones usually for manufacturing and safety aspects. Instead, they prefer to develop through an evolutionary process, for example, aircraft configurations such as Boeing B-707 / 737 / 747 / 757 / 767, Douglas DC-9 /DC-10 / MD-80 / MD-11, and Airbus A300 / 310 / 320 / 340. Of course, revolutionary concepts become conventional and commonplace as time passes and as they became more widespread. This was the same case with both Jet propulsion and sweep-back wing. Today these are generally accepted as "Conventional Concepts" without any confusion.

Focusing on Conventional High Subsonic, Civil, and Jet Transport aircraft which are still in wide use, a broad investigation on current airliners brought to light the following trends from the diverse transport configurations shown in Figure 2.1.1/1 and Figure 2.1.1/2.

1. The Main Wing is of a backward - swept shape and is positioned at the low or high part of the center fuselage.

2. The Number of Engines is two (2), three (3), or four (4).

3. The Horizontal Tail is of a backward-swept shape and is positioned at the rear fuselage or at the top of Vertical Tail. The Vertical Tail is also backward swept and positioned at the rear part of the fuselage.

4. Most configurations were designed with a LOW WING. A Mid-Wing position, although satisfactory aerodynamically, was very difficult to find among airliners because the wing must pass through the cabin and accordingly obstruct the seating arrangements.

   Therefore, the notable tendency was in favour of a LOW WING, which also contributes to safety aspects of transoceanic Long Range Transport when the aircraft may land on the water surface in an emergency situation. Of course, Mid - Wing Configurations can be found in trainer and military aircraft.

2.1.2 Review of Unconventional Aircraft

Compared with "*Conventional*" aircraft which have main lifting surfaces near the center fuselage, tail surfaces near aft-fuselage, and the passenger / cargo in one fuselage, "*Unconventional*" aircraft can be regarded either as a one "off conventional" type or as one which contains any combination of twin fuselage, tail-less type, canard, three surfaces, or forward-swept wings, etc.

All these unconventional aircraft were stimulated by the designer's efforts to find more efficient configurations. Sometimes external influences such as the fuel crisis in early 1970's motivated the aircraft designer to consider unconventional aircraft design concepts to achieve particular performance objectives or operational improvements in effective drag reduction, increased useful load fraction and enhanced airfield performances. Thus a number of approaches were implemented to design fuel efficient aircraft and some of these resulting in very large aircraft (VLA) for air cargo and variable & fixed geometry designs for 200 to 400 passenger sized transport. Other configurations worthy of paying attention to are oblique wing, ring wing, tilt rotor, X-wing, scissor wing with twin fuselage, multi-body cargo carrier, etc. The figure 2.1.2/1 shows some examples of unconventional aircraft. [26] [45]

2.1.3 Review of "State of the Art" Technology

Since conventional aircraft benefited from advanced technologies such as the Supercritical Wing, Mission Adaptive Wing (or Variable Cambered Wing), Advanced Composite Materials, Advanced Turbofan, Propfan Propulsion unit and Laminar Flow Control, these technologies were also incorporated into unconventional configurations.

A review of the trends of configuration transition showed that many configuration variations were closely coupled with the technological advances.

Some of these technologies were incorporated without hesitation as they were believed to be within a minimum degree of safety. Others, however, were adopted at the design commencement on the assumption that the technologies involved would have a certain potentiality of being fully developed at least before design completion. Otherwise, they are incorporated through modification in later stages of operation. [45]

This is what is called the *"State of the Art" technology*. It is important to have the knowledge about this as it will certainly influence aircraft weight, aircraft life cycle cost, prolonged operation, and service life, even if the cost of applying the "State of the Art" technologies seems to be expensive at the start of design.

Some of those, as summarized in Figure 2.1.3/1, were incorporated into the present Expert System. Others are worth mentioning for a possible extension in a later stage, even though they were not programmed into the present system partly due to Expert System capacity and partly due to highly sophisticated advances at this time.

2.2 Aircraft Design : Its Knowledge and Problems

Aircraft design, in its nature, requires a wide spectrum of knowledge from a variety of sources combined with designer's experience and judgement, as shown in Figure 2.2/1. Even in case of a wing design [1], the designer requires plenty of design knowledge to create a wing configuration that satisfies its requirements. Moreover, there are many different configurations, as shown in Figure 2.1.1/1. Thus the knowledge for aircraft configuration design is **"voluminous"**.

Certain knowledge can be described as hard in that it is explicitly mathematical in form (i.e., Lift to Drag ratio) while others (i.e., Experience and Judgements) are less precise usually with no mathematical characteristics and are described as soft. Aircraft design knowledge, hence, possess both **"hard and soft"** characteristics.

In the past, the aircraft designers had to incorporate "Metal Structure" only into a whole configuration design. At present, however, the usage of composite metal is widespread and must be included as a design material. Also, conventional wing contour is being shifted to "Supercritical Wing" (i.e., RAE Supercritical Airfoil series) with respect to high subsonic Jet Transport and the "Variable Cambered Wing" is also being developed. Hence, aircraft design knowledge is **"constantly changing"**, as technology advances.

With this *'voluminous', 'hard and soft', and 'constantly changing'* knowledge, the designer's task is to solve aircraft design problems, searching with this knowledge for a successful configuration. However, the task is extremely difficult because the designer cannot easily simplify what is a 'multi-facet' problem which has the following characteristics.

- The aircraft configuration design problem is **'Multidisciplinary, Interdisciplinary, and needs Compromise'** in that design activity involves not only the integration of major components but also the application of aerodynamics, propulsion, flight control, structure and material, avionics and its associated subsystem, blended in an effective way to describe an efficient configuration, as shown in Figure 2.2/2. *Otherwise it will result in one of "Undesirable as well as Humorous" configurations as shown in Figure 2.2/3.* [5] [38]

- The problem is also of **"multi-phase and multi-process"**, as shown in Figure 2.2/4. Aircraft design phases can be divided largely into 'Conceptual, Preliminary, and Detail' Design Phases. Further, preliminary design can be divided into parametric study, wing design, fuselage design, engine design (or selection), tail design, etc. Parametric study can be also divided into the processes to find wing loading, thrust loading, and cruise conditions.

- The problem possesses, in nature, **multiple measures of merits** for judging the goodness of design. It is difficult to answer the question of "Which is the most successful configuration design among 'similarly configured' aircraft such as B - 747, DC-10, and A-300 series ?", because they have their own merits and demerits.

In conclusion, to solve the design complexity and to search for an efficient configuration that satisfies requirements, designers must have the ability to

1. *Represent many types of design knowledge* from wide sources and to understand *its relationships*.

2. Understand and *structure design problems*.

3. Reason *"How to Search for a solution ?"* with the represented knowledge as above.

2.2.1 Aircraft Configuration Design Knowledge

As shown in Figure 2.2.1/1, aircraft configuration design describes a certain configuration that integrates all the required components and that satisfies the requirements. The designer therefore must have the knowledge with respect to specification and airworthiness requirements, because they stipulate a type of aircraft(i.e., civil subsonic jet transport) and regulate its operational standards (i.e., climb angle with One Engine Inoperative(OEI) after take-off).

Aircraft components are generally 'lifting surface', 'fuselage', 'engine', 'tail', and 'undercarriage'. To design them, the designer must have the knowledge of each component's configuration, with respect to *"What shape does each component have ? (*TYPE*)", "How many of it are required ? (*NUMBER*)", and "Where can it be positioned for integration ? (*POSITION*)"*.

To design each component, the designer must have the numerical knowledge from aeronautical engineering disciplines required for its design activity. For example, wing design is to select an airfoil, to decide sweep angle and 3-dimensional shape, and to predict wing weight, in consideration of aerodynamics, aeroelasticity, and structural aspects.

Most of the above design knowledge can be represented and connected by the relationships such as rules, mathematical formulations, or heuristics. As will be described later, wing position influences engine position and the latter also influences horizontal tail position. Thus, an example of a typical rule is "If 'wing is at high wing position' and 'engine is underwing-mounted', then 'horizontal tail is vertical tail mounted'". In fuselage design, after deciding fuselage diameter and its length, the fineness ratio can then be calculated (Mathematical Formulation). In engine position, the number of engines can be 2, 3, or 4. If 3 engines should be installed, their probable position is either '*all rear fuselage mounted*' or '*2 wing-mounted engines with 1 engine rear-fuselage mounted*' (Heuristics). The designer therefore must know the relationships among the design knowledge.

Conflict often occurs, for example, if the fuselage diameter is to be widened for the better passenger comfort, this increases the total weight and causes large drag, thus requiring more thrust and bigger engines. To cope with such conflicting requirements, the designer must possess the knowledge of 'How to use design knowledge to search for a solution ?', based upon experience, judgement, predictable design trends, use of 'state of the art technology', flair, etc.

2.2.2 Type of Aircraft Design Knowledge

As shown in Figure 2.2/1, an aircraft designer's knowledge comes from a variety of sources such as specification requirements, airworthiness requirements (BCAR, FAR, JAR, etc), aeronautical engineering disciplines (aerodynamics, structure analysis, stability and control, performance, etc.), design characteristics of each component, design processes, experience and flair, established data, design trend, the "State of the Art" technologies, etc.

Some of this knowledge is codified and fixed (i.e., specification and requirement), broken down into subproblems (i.e., performance ---> landing, take-off, and cruise performance), and established in a data base (i.e., RAE and NACA Airfoil series), while others are judgemental (i.e., incorporation of the "State of the Art" technology), imaginative (i.e., unconventional configuration), subjective (i.e., designer's flair in favour of low wing position), and predictable (i.e., design trend in the usage of composite materials).

Thus, aircraft design knowledge can be classified into the following several types as shown in Figure 2.2.2/1.

- **Governing Knowledge** : This kind of knowledge is obtainable from specification and airworthiness (JAR, BCAR, FAR, MIL-SPEC, etc). It is also obtainable from mathematical formulations with respect to aerodynamics, structural analysis, etc. Considering the fact that these requirements must be adhered to by the designer, this governs the overall design processes and can act as constraint or criteria during the search for a solution.

- **Configuration Knowledge** : This knowledge relates the aircraft to its diversified configurations and relates configuration components to their type, number, and position for integration. Some of these configurations are existing, obtainable from data base, and proven to be safe in their usage, while others are under investigation, non-existing, or imaginative.

- **Design process and its decomposition** : It is always convenient to reduce a design task to a set of more simple ones. For example, performance process can be decomposed into subprocesses such as landing, take-off, and cruise. Then, it leads the designer not only to understand easily the type of design problem but also to reason about the ways the performance problem can be solved. Thus, the knowledge is quite helpful to cope with the complexity of the design problem.

- **Supportable Knowledge** : To obtain the knowledge needed for designer's judgement, recourse can be made to the established data through retrieval, results from a program execution, and predictable design trends. Otherwise, the designer can depend upon his experience, common sense, or sometimes flair. This knowledge is simple and acquired without undue difficulty. However, subjective characteristics come from experience and flair.

- **Metalevel Knowledge** : This is also called "Metaknowledge" which means knowledge about knowledge. Once factual design knowledge and its logical connections are represented, the methods to use and control such knowledge are needed to search for the solution. Therefore, it is required to

  1. set the priorities (or order of precedence) among the required considerations (i.e., low operating cost or comfortability), among the major process and its subprocess (i.e., fuselage design, wing design, type number, position, etc.), and among the knowledge sources (i.e., use common sense first, or leave it to designer's flair, when required knowledge is not available).

  2. guide the search processes and control the knowledge to be added to or excluded from the knowledge base by using heuristics or rules, which can effectively limit the search for the solution.

2.2.3 Configuration Design Problem

As shown in Figure 2.1.1/1, there are numerous types of configurations existing among civil subsonic jet transport. They can be classified into some configuration trends as shown in Figure 2.2.3/1.

The trends show that wing position and the number of engines influence not only the engine position but also the tail position. The problem in aircraft configuration design is searching for a successful type with the knowledge stated in the previous section. Thus the problem inevitably requires the designer, firstly, to conceive and classify the type, number, position of each component that can be conceived. Secondly, it is required to implement the following design activities, to describe the dimensions of the integrated configuration, as shown in Figure 2.2.3/2.

- Synthesis of all the required components for the overall configuration
- Sizing each component and analysis
  : wing, fuselage, engine, tail, undercarriage, weight analysis, and cost analysis
- Trade-off's for resolving conflict requirements, thus leading to a successful configuration.

2.2.4 Solution Strategy : Reasoning

It requires a huge amount of time for a designer to search for a desired configuration by examining all the configuration types one by one, because even a configuration design needs time-consuming, numerous, and repetitive trade-off's among design processes.

Thus, as Figure 2.2.3/1 shows, the designer needs a special 'solution strategy' to efficiently control the search for a configuration solution and the strategy requires human reasoning. This *human reasoning* is an activity, which is still poorly understood and traditional computer techniques can not simulate.

Traditional computer techniques can support the designer only in specific areas such as numerical calculation and data processing. However, within computer science, researchers of Artificial Intelligence (A. I.) have made vigorous efforts, since mid 1960's, to build computational model which can simulate human reasoning and could be regarded as *'intelligent'*. One of their results is a system, which is now called an 'Expert System'. [51, 52, 53, 56, 57, 72, 79]

The Expert System provides useful benefits in limited but difficult real world problem domains in that

1. It helps not only in computer-related tasks such as numerical calculation or information retrieval but also in the tasks which require human reasoning.
2. It simulates the reasoning process of a human expert. Hence it provides a methodology of 'how to use expertise in specific problem domain'.
3. It makes the knowledge transferable by storing expertise in 'Knowledge Base'.

4. It not only has representation of knowledge dealing with problem domain but also maintains its own representation. It's self-representation called explanation function explains the process of 'how a design problem could be solved'.

2.3 The World of Artificial Intelligence

2.3.1 Introduction

It is very difficult to define Artificial Intelligence (A.I.). A precise definition of A. I. has not yet been formulated and the definition is still evolving. However, its definition can be interpreted from the following well-known Turing's Test.

-. An interrogator is separated from a person (or machine) under interrogation and communication is only possible using a Terminal. The idea is that if human cannot tell, through the interrogation, whether communication is with another person or a machine, then the machine - if indeed it is machine giving answers - may be regarded as intelligent. -- [Turing's test] [1] [58]

Since then, computer scientists became interested in building a computational model that could be regarded 'intelligent' in solving problems as if done by humans. This notion of "computer's intelligent solving" led A.I. researchers to represent knowledge in problem domain and to reason about these representations. [51]

With the energetic exploration by imaginative A.I. researchers, like Winograd, Minsky, Quillan, etc., there also appeared the knowledge representation schemes such as rule, semantic nets and frame(*). [54] [61]

The brilliant advances in fourth generation computer technique using silicon chips and the researches between the human reasoning and human intelligence, led to some applications to real world problems together with the development of a program by using the new languages such as LISP, SMALLTALK, and PROLOG. [53, 73, 77, 79, 81, 86]

(*) : The explanation of these terminologies were well described in reference. [61]

Also it is particularly worth noting that the fifth generation computer project in Japan has been commenced with PROLOG, for the purpose of developing the, literally, Artificial Intelligence Computer. [56] [80]

While the work progressed, it was realised by A.I. researchers that "general intelligence" was indefinable. Accordingly, they used the domain specific knowledge that humans possess, in tackling in detail the real world problems.

This distinguishes the present day A.I. work from the conventional problem-solving programs as well as from the previous A.I. work. The present A.I. researches work with systems which contain the following characteristics.

- The knowledge is domain specific and represented in, what is generally called the "knowledge base".
- The reasoning is performed by a Control Mechanism, which is called the "Inference Engine".

These characteristics negate the necessity to make changes to both parts together. It means that one might only add knowledge to a "knowledge base" without altering the inference engine, and vice versa also.

The work has given rise to a number of applications with a high level of performance on non-trivial tasks such as MYCIN for treating blood infections, X1 for configuring VAX range of computer, MECHO for solving Newtonian Mechanics problems, etc. [1, 51, 52, 56, 61, 62, 79, 87]

2.3.2 The Knowledge Base for A.I. Development

The A.I. Technique uses basically knowledge and therefore it concentrates on Knowledge Engineering, where knowledge means an expertise in a given area (which is domain specific and could be a highly technical fact or a domain of heuristic function, e.g., knowing rules of practice and plausible reasoning and possessing judgement) and the engineering means utilizing such knowledge to tackle problems such as treating blood disease, mining gold, decision making, engineering design, and so on.

The system developed in this way is called a "Knowledge Based Expert System" or a "Rule Based Expert System", because these systems uses the knowledge of an expert (in other word, expertise) and the rule to process knowledge for plausible reasoning.

So far, in the process of developing the A.I. techniques, the following areas contributed a great deal to forming a good knowledge base for A.I. development. [55]

1. *Mathematical Logic*, which was developed in the 1930's and the 1940's and enabled computation to process with symbols.

2. *Psychology* investigated the dimensions of a human's ways of thinking, reasoning, and intelligent behaviour.

3. *Cybernetics*, which is concerned with control processes in electronic, mechanical, and biological systems, examines the flow of information within a system and the schemes to control the flow of information.

4. *Predicate Calculus*, a branch of mathematics, uses assumptions and axioms to prove propositions.

5. *Semantics, Syntax, and Lexicon of human languages* are good means for structuring the A.I. programs in the areas of natural and human language.

6. *Vision* is for investigating how people identify and recognize a thing.

2.3.3 A.I. Application Areas

The A.I. application fields are very extensive and they include problem-solving, perception, natural language, learning and induction, robotics, and expert system. For this research, the author mainly paid attention to problem solving and expert systems to get lessons applicable to aircraft design.

2.3.3.1 Problem Solving

Game playing, theorem proving, and general problem solving fall within this scope.

The idea called the "State Space Search" considers problems in terms of a starting state, a final state, and a set of operations to achieve the final state in a solution space. The set of operations (i.e., legal chess move) includes *'generate and test', depth-first search, breadth-first search, best-first search, and heuristic* *search* which uses evaluation functions together with reasoning and narrowing down to the goal using constraints. [51]

As indicated, the method derived from this approach can be applied to real world problems such as game playing, theorem proving, decision-making in management systems, economic problems, and social problems, although the human environment is constantly changing and difficult to define.

2.3.3.2 Expert System

It was the goal of A.I. scientists to develop computer programs that could, in some sense, think, and solve problems intelligently as if done by humans.

As the trend shift of the A.I. shows in the Figure 2.3.3.2/1, A.I. scientists began to realise that a source of problem solving power comes from high-skilled and specific knowledge in problem areas rather than a formalism and reasoning scheme which a problem possesses.

Thus, instead of developing a general purpose program which was extremely difficult and not very useful, they concentrated on developing a special purpose system with which the user can solve domain specific problem as if solving through interacting with an expert. This made it possible to develop a special-purpose computer program, expert in specific domains, which is now called " Expert System". [61]

Accordingly, the Expert Systems perform in a manner similar to a human expert possessing a domain specific expertise. For example, to cope with the emergent case that a certain problem must be solved without a domain expert, plenty of experiences of the human expert can be programmed and stored in an Expert System. Thus much progress has been made in the development of Expert Systems, some of which are known to outperform human experts (e.g. MYCIN). [56] [79]

1. The Characteristics of Expert Systems

The Expert System separates the knowledge of a problem domain called the " *knowledge base* ", from the knowledge of how to solve the problem and to interact with the user, the latter is called the " *Inference Engine* ". Thus, the program organised in this way is called the "Knowledge Base System". The Figure 2.3.3.2/2 can illustrate the relations among A.I. programs, the knowledge-based systems, and the Expert Systems.

The major components of Expert Systems consist of the user interface, organisation of knowledge, inference, and search method as shown in the Figure 2.3.3.2/3. Then the Expert System must, at least, exhibit the following characteristics. [55] [79] [80]

-. possess **Expertise** with an expert's level of performance and high level of skill.

-. possess **Symbolic Reasoning** with a symbolic knowledge representation and a reformulation capability if required.

-. possess **Deep Knowledge** capable of handling a difficult problem and using complex rules.

-. possess **Self-Knowledge** with a capability to explain the reasoning process (How and Why ?) and operation.

The above characteristics make the Expert System different from conventional software programs as shown in the Table 2.3.3.2/1.

Table 2.3.3.2/1 Difference : Conventional Software and Expert System [61]

Conventional Software	*Expert System*
Data Base : Representation and use of data	Knowledge Base : Representation / use of knowledge
Procedural and Algorithmic	Heuristic and Rule of Thumb
Interpreter / Compiler with repetitive process	Inference Engine with inferential process
Very effective in handling large data bases	Very powerful in handling large knowledge bases

2. The Evolution of Expert System

As the Expert Systems have been developed and progressed, many advances have evolved, which were based upon the previous " naive concepts " and former systems.

Each original system for a specific field produced derivatives with new characteristics and new methodologies. For example, the DENDRAL PROJECT [79], which was developed at the Stanford University in the late 1960's for inferencing the plausible structure of an unknown chemical compound, produced both DENDRAL and META-DENDRAL. [51] [56]

Likewise, the SAINT, developed at MIT in 1961 for performing differential and integral calculus, resulted in MACSYMA. Similarly, the CASNET, developed for consultation in the diagnosis and glaucoma, led to EXPERT (Expert System building language) and MYCIN, developed for diagnosing blood infection, gave rise to the EMYCIN & PUFF, TEIRESIAS for knowledge base construction, PROSPECTOR for a mineral deposit, KAS, and RITA & ROSIE for general purpose programming systems. [56] [61]

PSG, developed at the Carnegie-Mellon University in 1973, for modelling human cognition, led to the OPS series for the production system languages and R1 for configuring the DEC VAX Computers. [61] [62]

In a similar manner, the HEARSAY-II system, emanated from the HEARSAY-I and developed at the Carnegie-Mellon University in 1980 for speech understanding, led to the HEARSAY-III & AGE for developing general purpose frameworks to build an Expert System based upon the HEARSAY-II. [79]

3. Functions of Expert Systems

The activity types of problems that Expert Systems can solve can be described as follows; [61] [79] [80]

CATEGORY	PROBLEM DESCRIPTION
Interpretation	Inferring situation description from sensor data

Prediction	Inferring likely consequences of given situations
Diagnosis	Inferring system malfunctions from observables
Design	**Configuring objects under constraints**
Planning	Designing actions
Monitoring	Comparing observations to expected outcomes
Debugging	Prescribing remedies for malfunctions
Repair	Executing plans to administer prescribed remedies
Instructions	Diagnosing, debugging, and repairing student behaviour
Control	Governing overall system behaviour

4. Expert System Building Phases

The build - up of Expert Systems can have the phases such as **Identification, Conceptualisation, Formalisation, Implementation,** and **Testing**, as shown in the Figure 2.3.3.2/4. [61] [80]

1. Identification

This process is for the knowledge engineer and domain experts to identify the problem type and scope, participants / additional experts to be involved, the required time, their associated facilities, and the goals and objects to be pursued in building an Expert System.

2. Conceptualisation

All the concepts, relations, and control mechanisms must be decided, together with subtasks, strategies, and constraints related to problem - solving.

3. **Formalisation**

The key concepts and relations are to be expressed in a formal way within a framework and an expert system building language, which means the selections between the approaches such as rule-based approach (e.g., ROSIE) and frame-based approach (e.g., SRL).

4. **Implementation**

This process turns the knowledges, formalised in the process 3, into a computer program.

5. **Testing**

The program developed in the process 4 is to be evaluated as for the performance and utility of a prototype and it is accordingly revised.

2.3.3.3 Knowledge Representation Techniques

1. Rules

The rules are expressed as 'If Then' statement as follows;

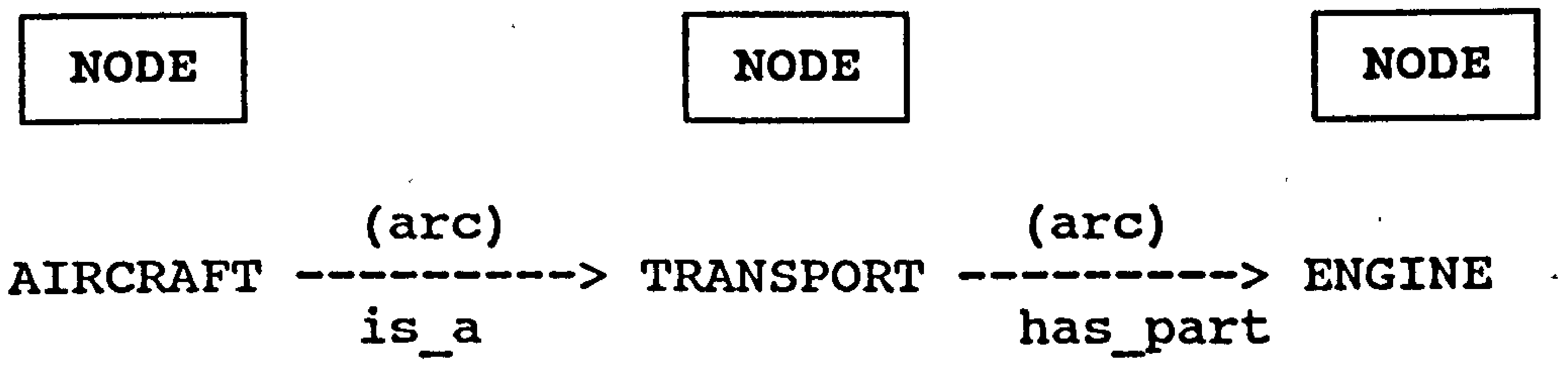
- If a wing is at low-wing position, then the engine can be mounted under the wing or at the rear fuselage.
- If a engine is mounted at the rear fuselage, then the horizontal stabilizer is mounted at the fin.

These rules can be used for determining the configuration of subsonic transports. In the "if-then" rules, "then" can also be replaced with an "arrow (------>)". When the "if" statement is satisfied, the "then" portion is performed. This activity is called the *"fire rules" or "execute rules"*. [80]

2. Semantic Nets

The Semantic Net is sometimes called the "Semantic Network". Its components consist of arcs, nodes, hierarchies (is_a, has_part). [61]

Sentence : One kind of aircraft is transport and the transport has a part called the "engine".



3. Frames

It is still an open question among A. I. researchers what is the best method to represent a knowledge. Frame system can be regarded as one way of circumventing the demerits of "Rule Based System" and advancing the modelling of real world systems. It is one kind of template for holding the clusters of relevant knowledge about a particular subjects.

The frame was originated by Marvin Minsky and could be described as follows. [61]

"A frame is a data-structure for representing a stereotyped situation, like being in a certain kind of living room, or going to child's birthday party. Attached to each frame are several kinds of information. Some of this information is about how to use the frame. Some is about what one can expect to happen next. Some is about what to do if this expectation are not confirmed."

The frame system organizes a network of nodes and relations in a hierarchy and thus the frame-based system also includes the semantic nets and frames altogether.

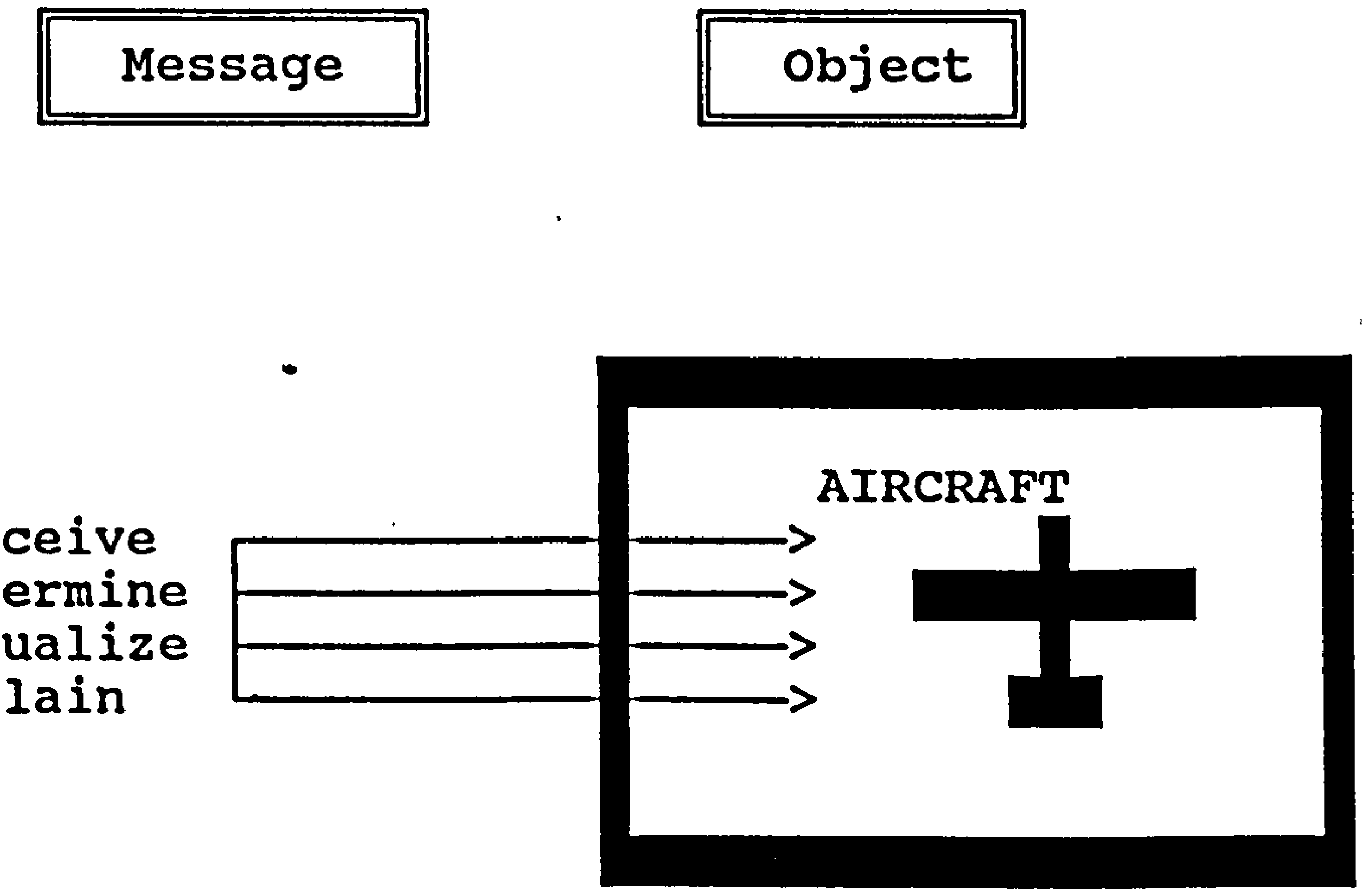
However, it is much more complicated to construct a knowledge system based upon a frame than to construct with a rule - based system. The Rule Based System is more handy in that it is easy to structure the simple, smaller - sized, and 'If-Then' formatted knowledges, while the 'rule' representation of interrelationships are excessive in some areas where rule, conditions, or attributes tend to cluster a central objects. [87]

Thus, it is the 'Frame Based Systems' that were developed to structure a knowledge by taking advantage of natural clustering not only in individual frames but also among many frames themselves.

4. Object Oriented Programming

In conventional programming, the programs can be considered as primary, while the data or object is secondary. This means that the program is given the objects to manipulate. Contrary to the Conventional Method, it is of prime importance in the object oriented method to think about, above all, the data or object to be operated. [80] [87]

The object receives a message which is the name of operation / procedure (called the 'Method') to perform. For example, 'Conceive, Determine, Visualize, Explain the configuration of aircraft' can be expressed as follows;



Further, if the objects can be classified hierarchically, a hierarchical inheritance is incorporated in the object- oriented programming, which means that the programmer does not have to change or add the system code due to an addition of new class.

2.3.3.4 A.I. Languages

A.I. languages are different from a numerical calculation language such as the FORTRAN. The following three symbolic processing languages have been developed and have been widely used for applications. [52]

At the time of developing this system, LISP was not available and the SMALLTALK was discarded as stated below. Thus PROLOG was adopted in consideration of its features.

1. LISP

LISP is, at the present time, more widely used in the U.S.A. than in EUROPE and is one of the most important members of the A.I. languages family, considering the number of lines of code written in it and its influence on developing the other languages. LISP, first presented in as a notion for defining mathematical functions, has become one of the favoured languages for A.I. system development. LISP is the abbreviation of LISt Processing and many programs have used it. [79] [80]

The key reasons why LISP is a good language for building A.I. systems are : [52]

1. Its principal data structure is composed of lists which are known to be very powerful for the knowledge representation in A.I. programs. The lists which represent a property of an object need not to be in a fixed - size and the program execution can change dramatically once the properties already known.

   The number of facts regarding an individual object can easily be represented in the property list, which is associated with the concept represented in the Atom and is simply the attribute - value list pair.

2. Recursion is used for the control structure, which is appropriate for many problem - solving tasks and is also used in the other A.I. language (e.g., PROLOG).

3. Both the data and procedures, represented as lists, enable the declarative and procedural knowledge to be integrated into a single structure such as

property lists and enable a program to construct / execute the procedure.

4. The interactive operation, which is very important for facilitating the development of many programs, can be implemented in LISP and is essential to the application field where problem-solving is impossible without human assistance and intervention.

2. PROLOG

PROLOG is an abbreviation of PROgramming in LOGic. It was based upon the predicate calculus and thus is a production rule language where programs are written as rules for proving relations among objects. PROLOG thus consists of a set of clauses and can also handle lists. We can express practically some useful features in PROLOG as follows; [53] [62] [72]

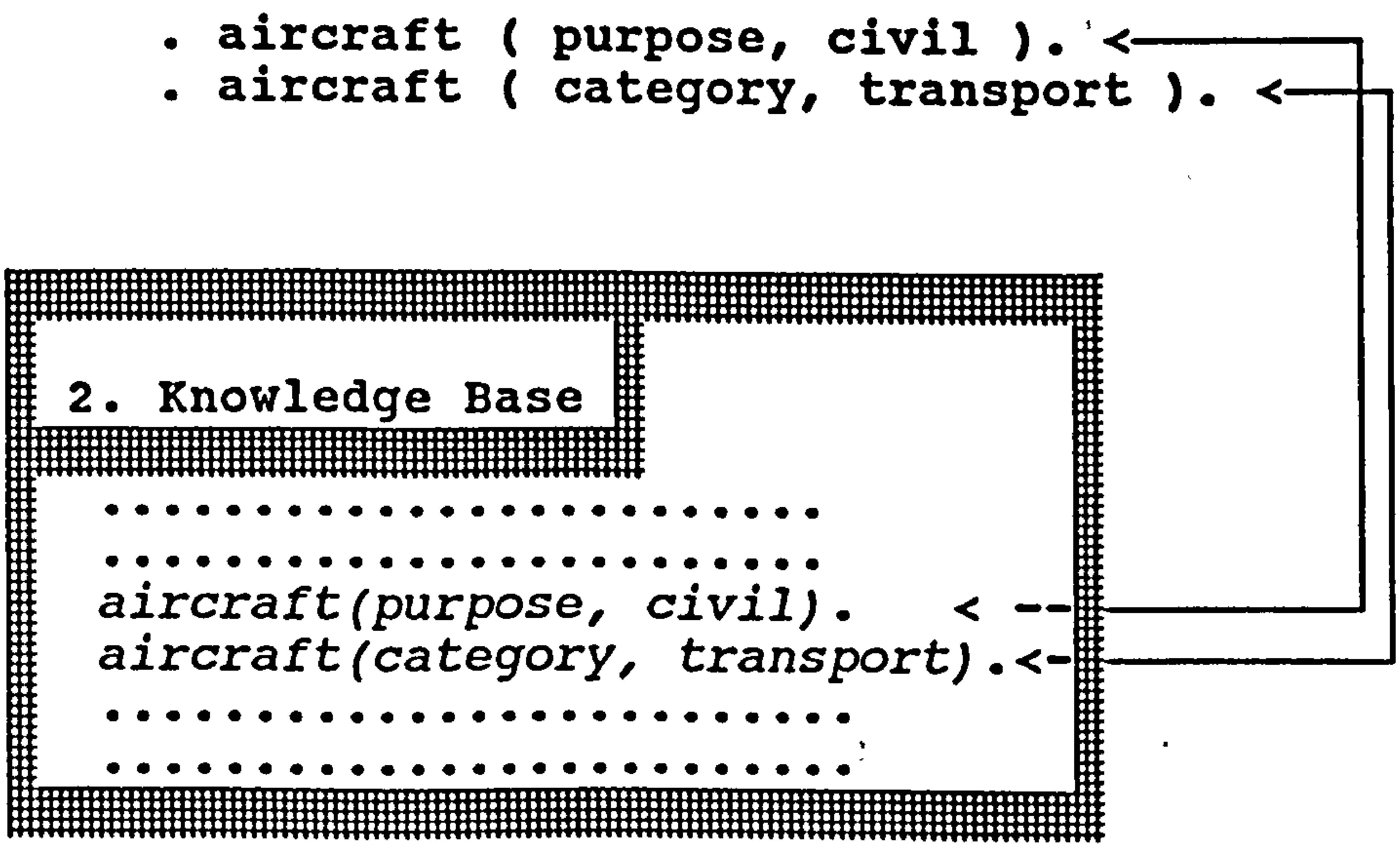
- The fact, rules, or any expressions can be represented in clausal forms as designer wishes to structure.

  e.g., Purpose of the aircraft is civil.
  Category of the aircraft is transport.

  ==> aircraft (purpose, civil).
  aircraft (category, transport).

- It is easy to find the desired facts through pattern matching in the form of a general unification.

  e.g., 1. The Desired Facts



- The built-in strategy (depth first search) which searches for the solution with the top of the data base being accessed first.

3. SMALLTALK

SMALLTALK is an object oriented programming language. It was developed primarily to explore and facilitate the high quality human machine graphics rather than being developed for A. I. developments. As an object- oriented language, SMALLTALK enables us to classify hierarchically all information into related characteristics. Thus, SMALLTALK makes it possible to break down problems into more manageable subproblems. [80] [87]

2.3.3.5 Inference Engine

By separating the knowledge base and the reasoning mechanism to support a particular problem, a better program for simulating a human reasoning can be constructed, which is an important characteristic of an Expert System. While the conventional procedural programs combine a knowledge and program control, the Expert System separates the knowledge and control structure, without the need to change both parts simultaneously even if one part is to be changed.

This inference engine is the heart of an expert system and simulates the reasoning process that people use for problem-solving activities. It performs the following important tasks.

- The Reasoning or Inference is based upon a system of formal logic similar to the predicate logic and uses simple "If-Then" rules to manipulate facts.

- The Inference Engine is responsible for determining the order in which the rules are selected to search for the solution and has two typical search methods , as stated below.

  A. The Backward-Chaining is a simple technique which starts with a conclusion and works backward through sub-goals to determine whether the conclusion is valid or not. This is better suited for the problems that have more start states than goal states.

B. The **Forward-Chaining** is the process which starts with facts and works forward and tries to find a valid conclusion. This is better for solving a problem having more goal states than start states.

The combination of these two control strategies can be used to add the more flexibility to an inference engine.

Item	A/C	Wing			Engine						Tail	
	Name	Low	Mid	High	2		3		4		R.F	F.M
No.					W	F	W	F	W	F		
1.	B757-200	O			O						O	
2.	B767-200	O			O						O	
3.	MD-80	O				O						O
4.	F - 28	O				O						O
5.	A320-200	O			O						O	
6.	L-1011	O					O	O			O	
7.	DC-10	O					O	O			O	
8.	B747-200	O							O		O	
9.	BAE-146			O					O			O

Note) R.F : *Rear Fuselage Mounted*
F.M : *Fin Mounted*
W : *Wing Mounted*
F : *Rear Fuselage Mounted*

Figure 2.1.1/1 The Configuration Examples Of Modern Transport Aircraft

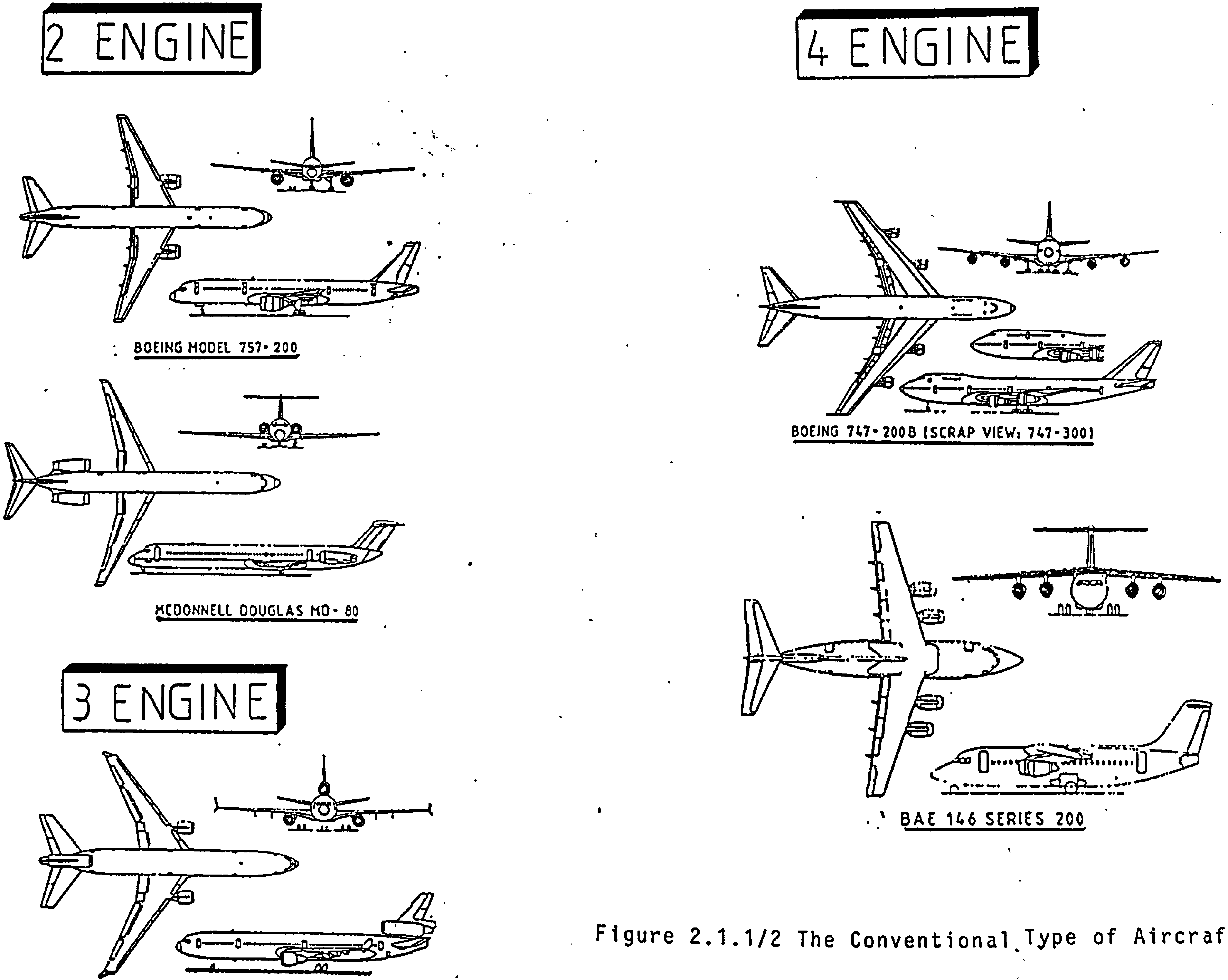
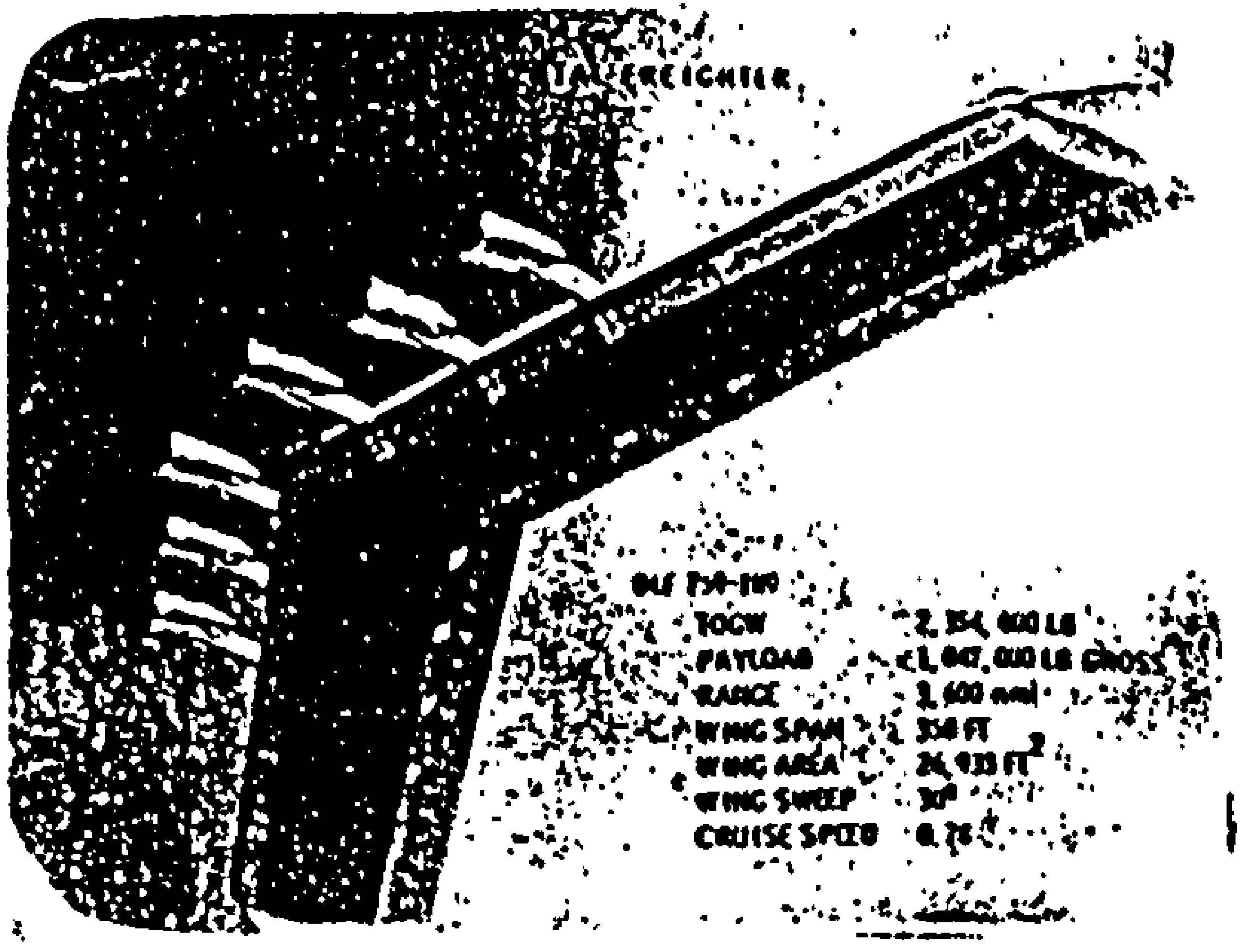


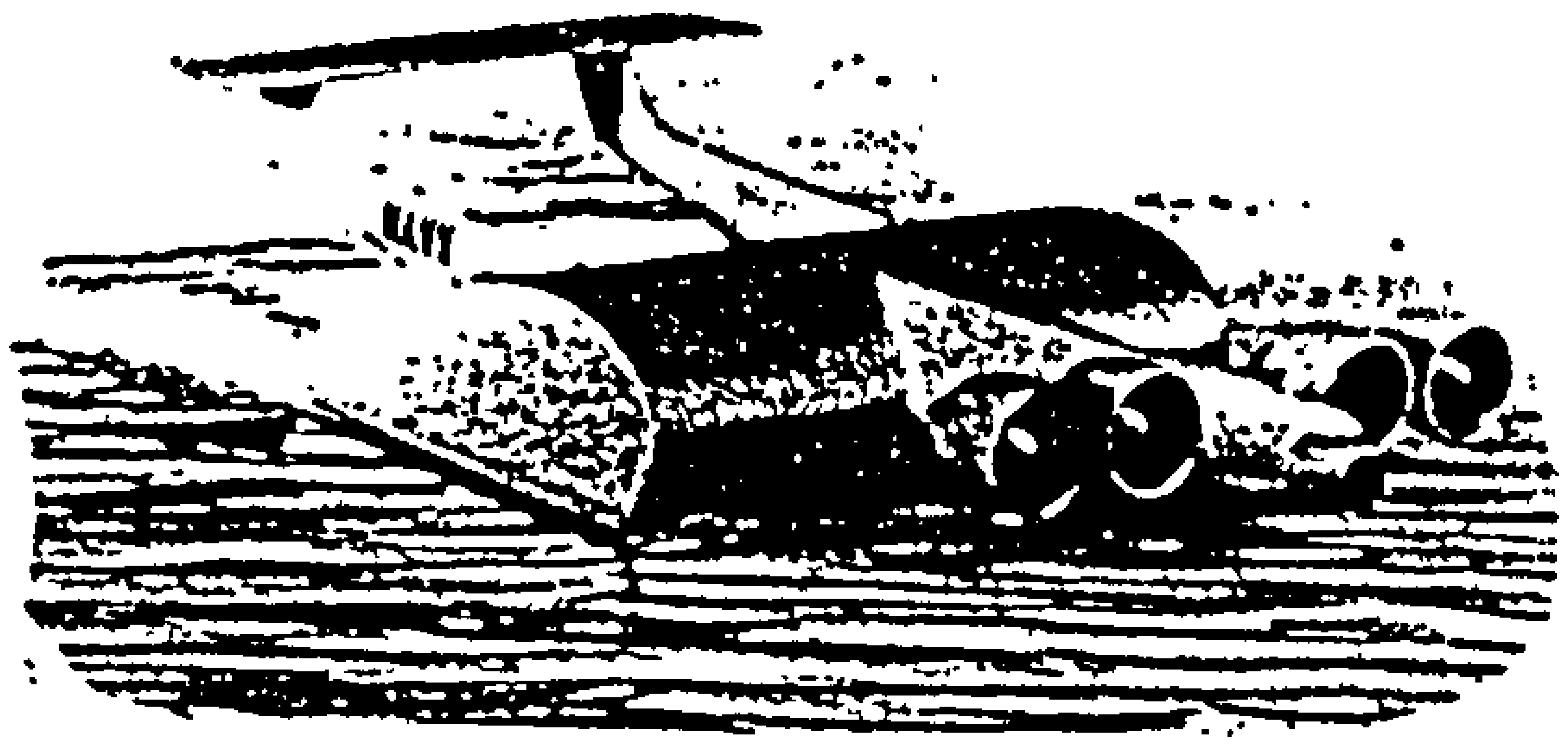
Figure 2.1.1/2 The Conventional Type of Aircraft

Lockheed Span Loader Concept

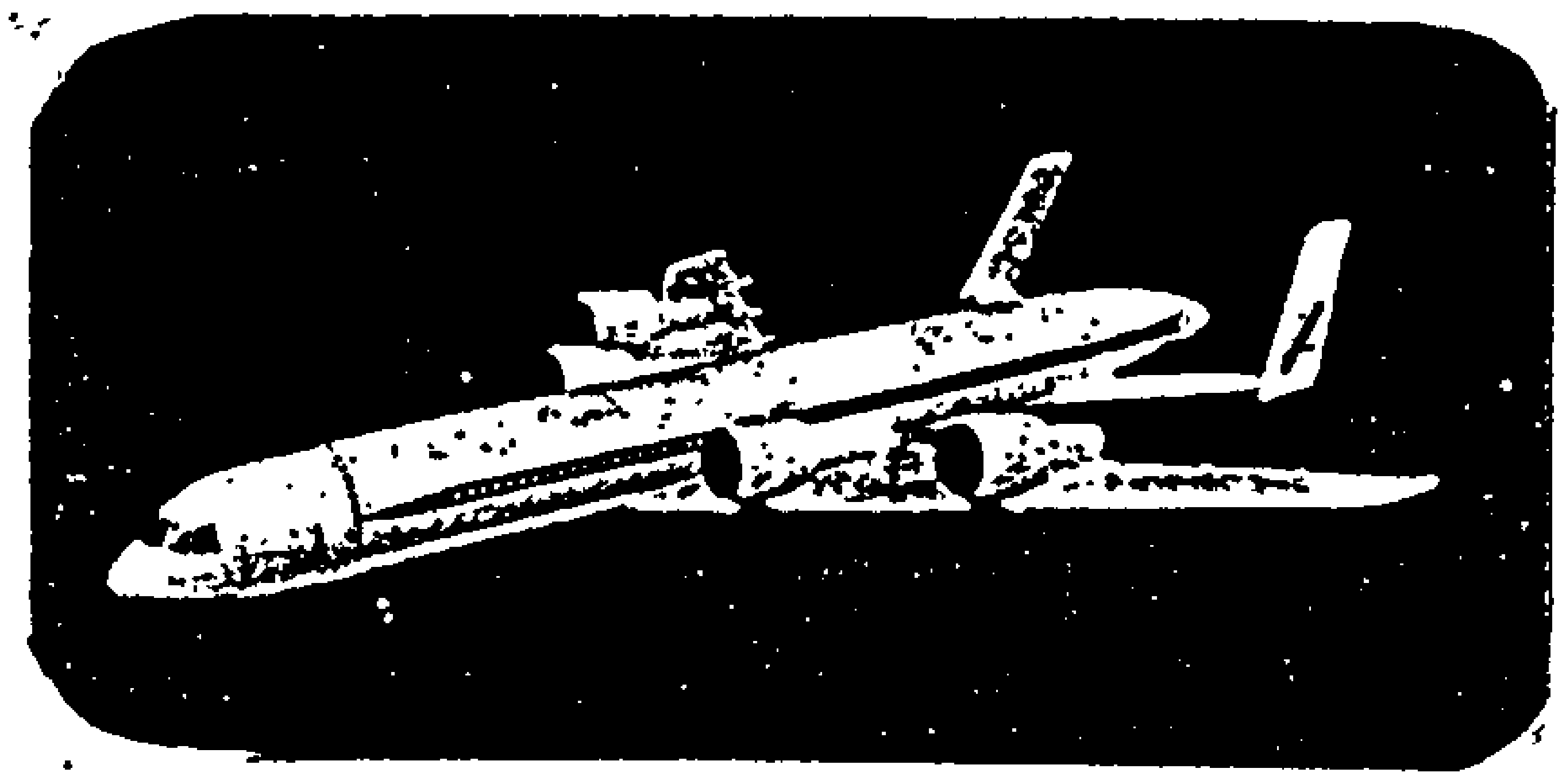


Boeing Distributed Load Freighter

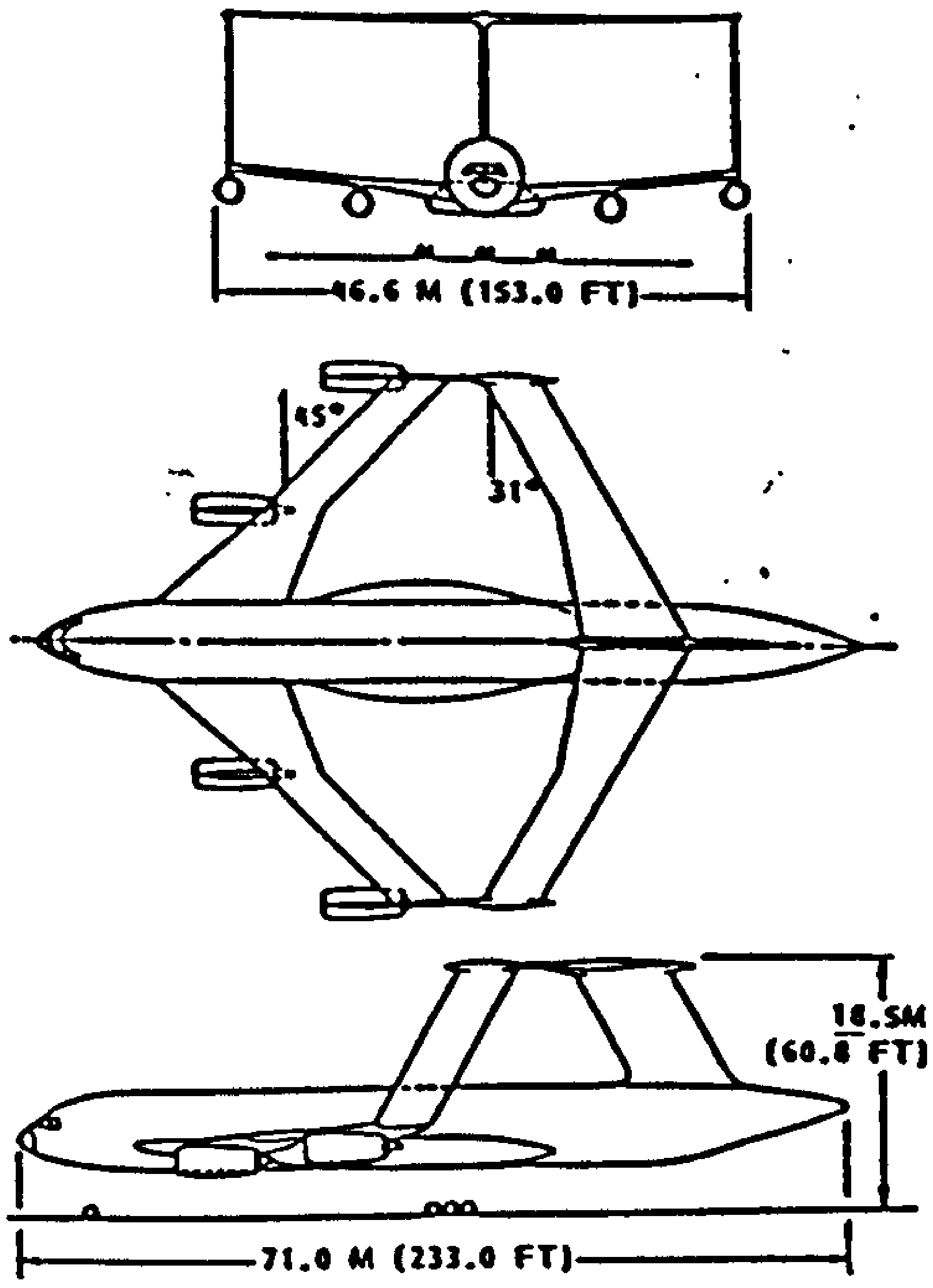
Multi-Body Cargo-Transport Concept



Wing-In-Ground Effect Transport

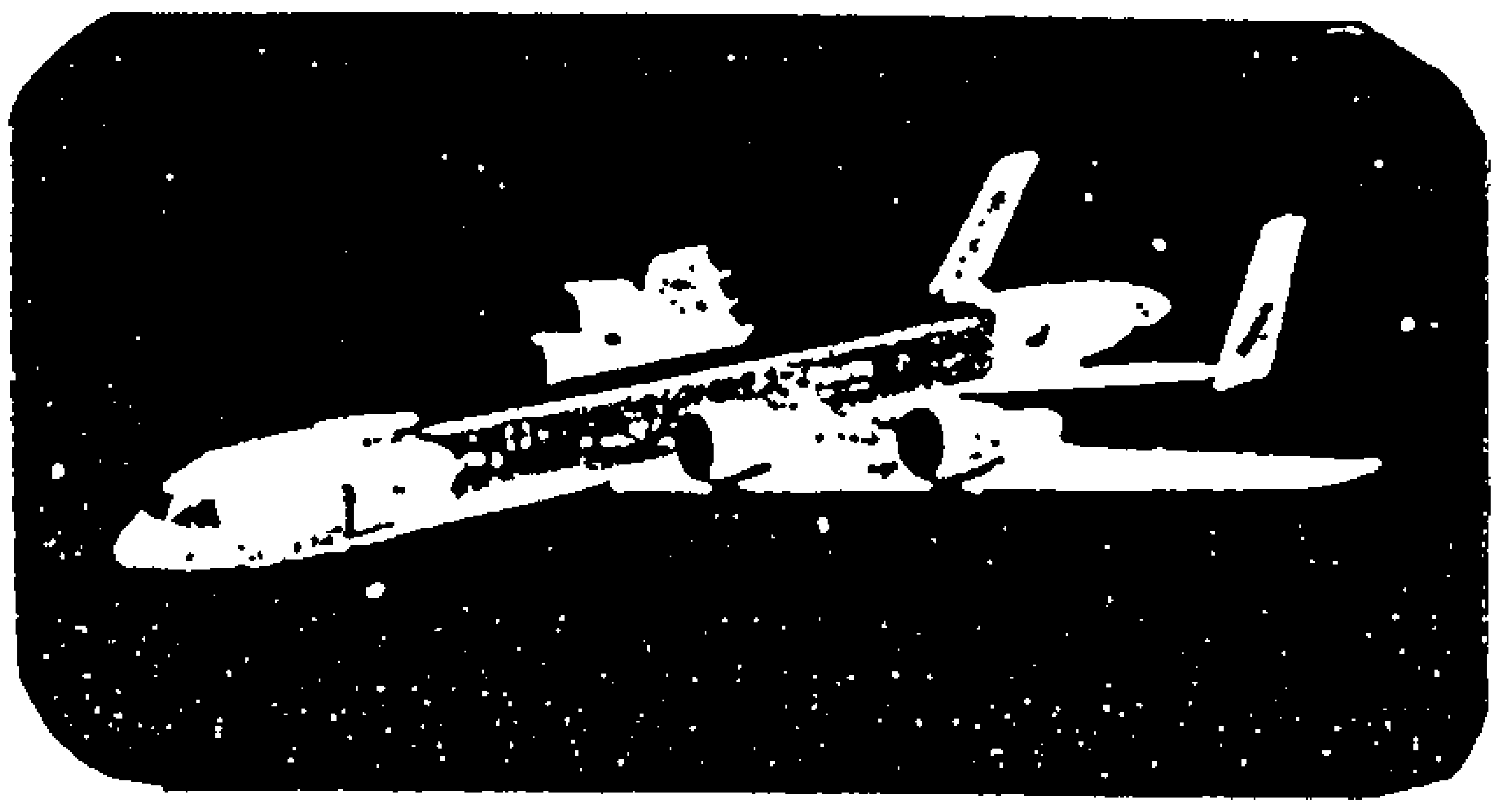


Flatbed with Passenger Module



SPEED	0.95
PAYLOAD	84,800 LB
RANGE	5500 NM
OPERATING WT	281,392 LB
GROSS WT	664,896 LB

Transonic Biplane Concept



Flatbed with Cargo Containers

2.1.2/1 The Unconventional Type of Aircraft [26]

DESIGN AREA	**STATE of THE ART TECHNOLOGY**	**MERIT/ DEMERITS**	**APPLIED AIRCRAFT**
WING	Forward Swept Wing	Avoid Tip Stall Higher Lift than Sweptback, thus less swept	Ju-287 HFB-320 X-29A
		Large bending due to wash in, heavy due to beep-up	
	Movable Forward (Canard)	Produce lift, light main wing Super Manoeuvrability	CW-24B X-10 Starship -I, SAAB Viggin
		Inherently not Stable in pitch	
	Supercritical	Delay shock Thick wing, thus more fuel volume ,Light Weight	Learjet B-747
	Winglet or Wingtip Turbine	Eliminate Wingtip Vortex, fuel efficient.	Gulf Stream PA-28
	Mission Adaptive Wing, or Variable Camber	Peak Aerodyn. efficiency	AFTI F-111
	Laminar Flow Control	Reduce boundary Layer , reduce drag fuel consump.	
		Weight Penalty Maintenance	
Propu-lsion	Propfan/UDF Unducted Fan	Less noise, Less drag, Less control, Fuel efficient,	MD-80
	Scramjet	Max. Speed over 3.5 up to 25	

		Device for starting to M_{cr} 6.	
	Plenum Chamber Burning Remote Augment Lift System (V/STOL Flight)	Accelerate over 0.9 up to supersonic speed	Pegasus
	Fuel, Jet A or LCH_4 or LH_2	Reasonable cost (except LH_4) Air Pollution	
Stability & Control	Fly-by-wire(or light)	Eliminate elec. mechan. link., so,less weight Simple & responsive control	F-16 F/A-18 Concorde
	Side-stick Controller	Convenient at right hand	F-16 A-320
	Active Control Technology	Auto. Control Avoid Flutter Reduce Bending	
	Control Configured Vehicle (CCV)	Super Manoeuvrability over stall angle	CCV-16 (F-16 Modified)
AVIONICS	Electronic Flight Instrumentation System (EFIS)	Small Space by flat panel display	
	Flight Management Computer System (FMCS)	Fuel Economy Efficient Operation	
MATERIALS	Al.-Li. Alloy Superalloy Composite	High Temperature application , reduce weight	

Figure 2.1.3/1 The List of the State of the Art Technology under study or experiment

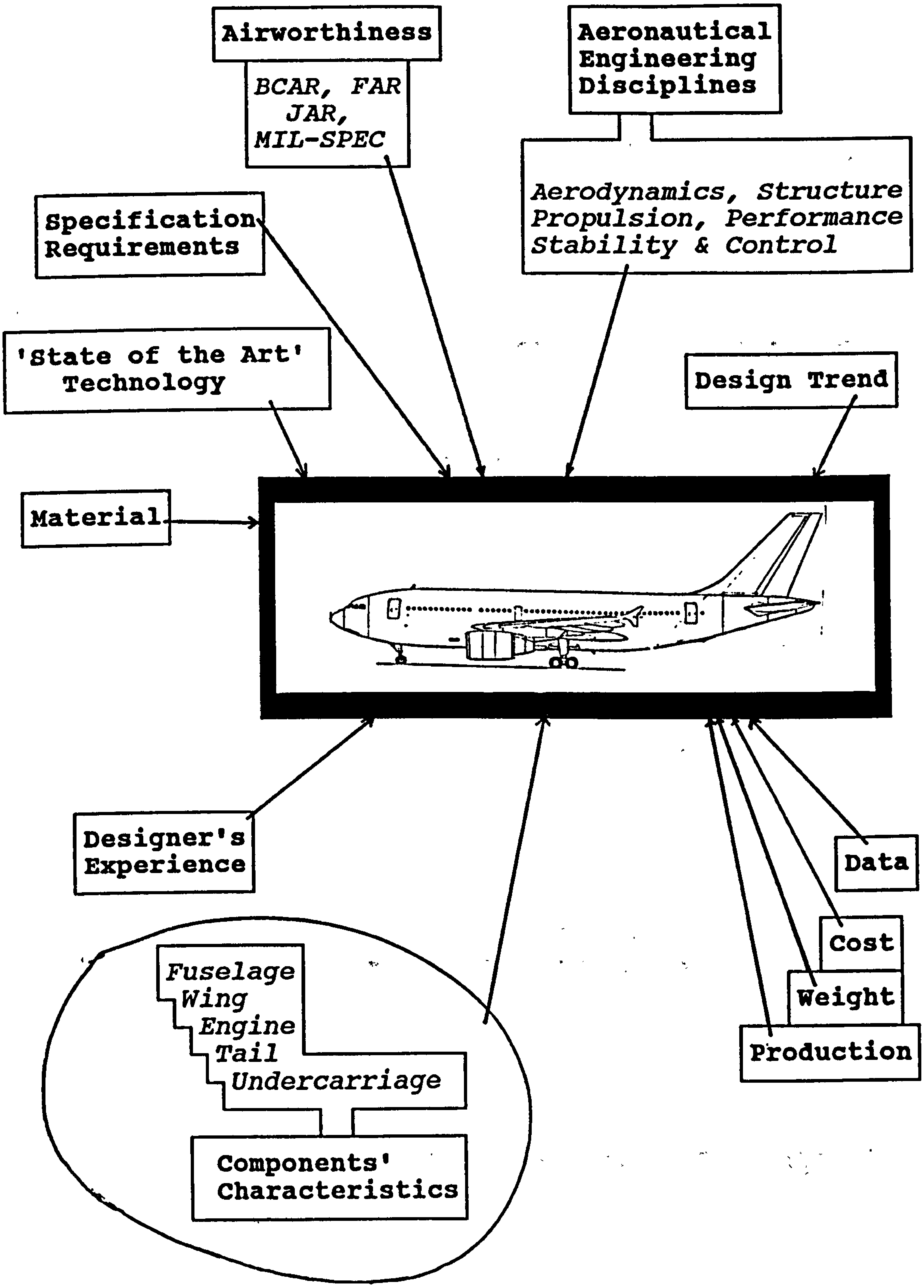


Figure 2.2/1 Aircraft Design Knowledge Sources

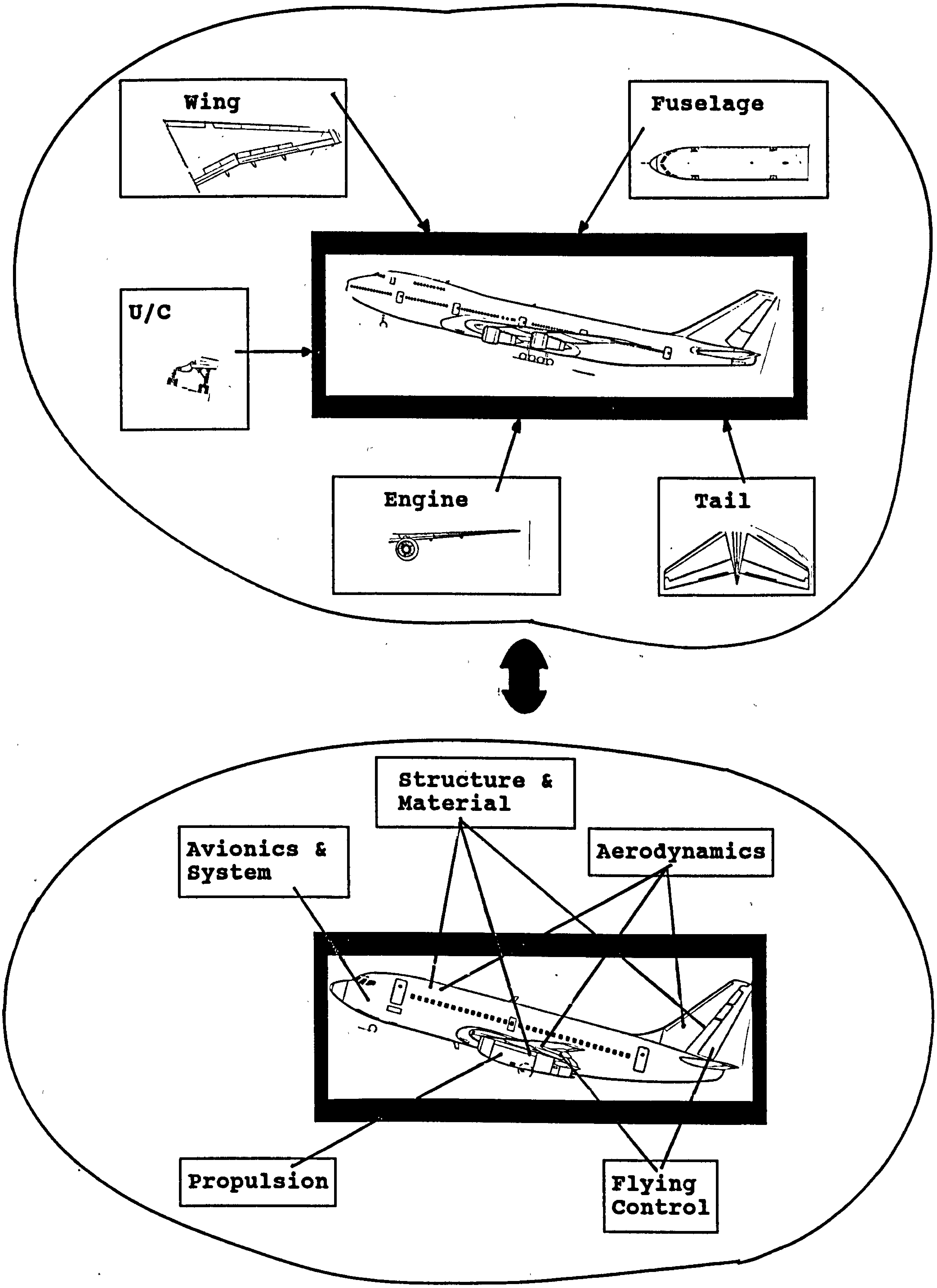
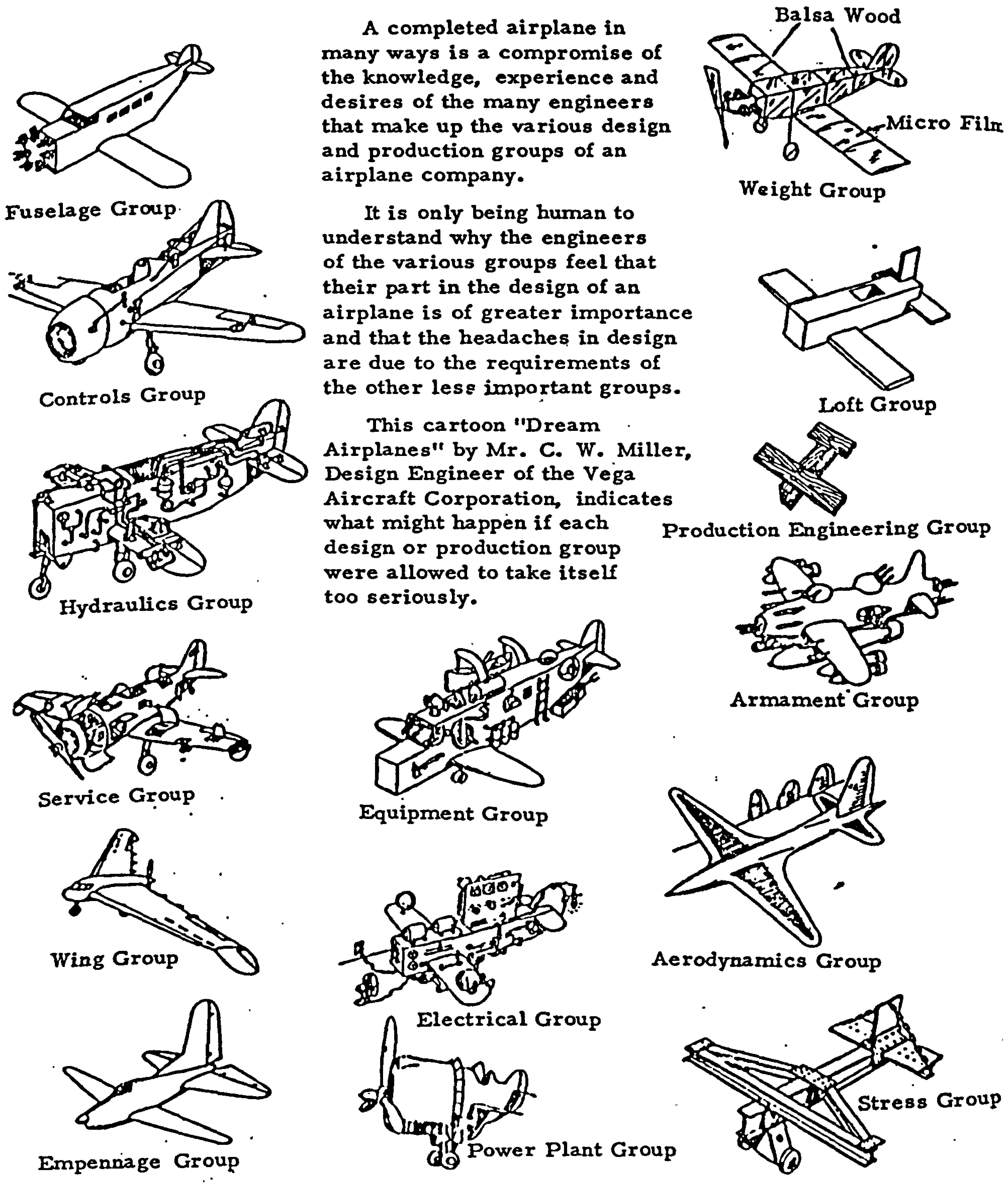


Figure 2.2/2 Integration of Major Components and Application of Disciplines



A completed airplane in many ways is a compromise of the knowledge, experience and desires of the many engineers that make up the various design and production groups of an airplane company.

It is only being human to understand why the engineers of the various groups feel that their part in the design of an airplane is of greater importance and that the headaches in design are due to the requirements of the other less important groups.

This cartoon "Dream Airplanes" by Mr. C. W. Miller, Design Engineer of the Vega Aircraft Corporation, indicates what might happen if each design or production group were allowed to take itself too seriously.

Figure 2.2/3 Undesirable (Aircraft) Configurations
(p 1-2, [5])

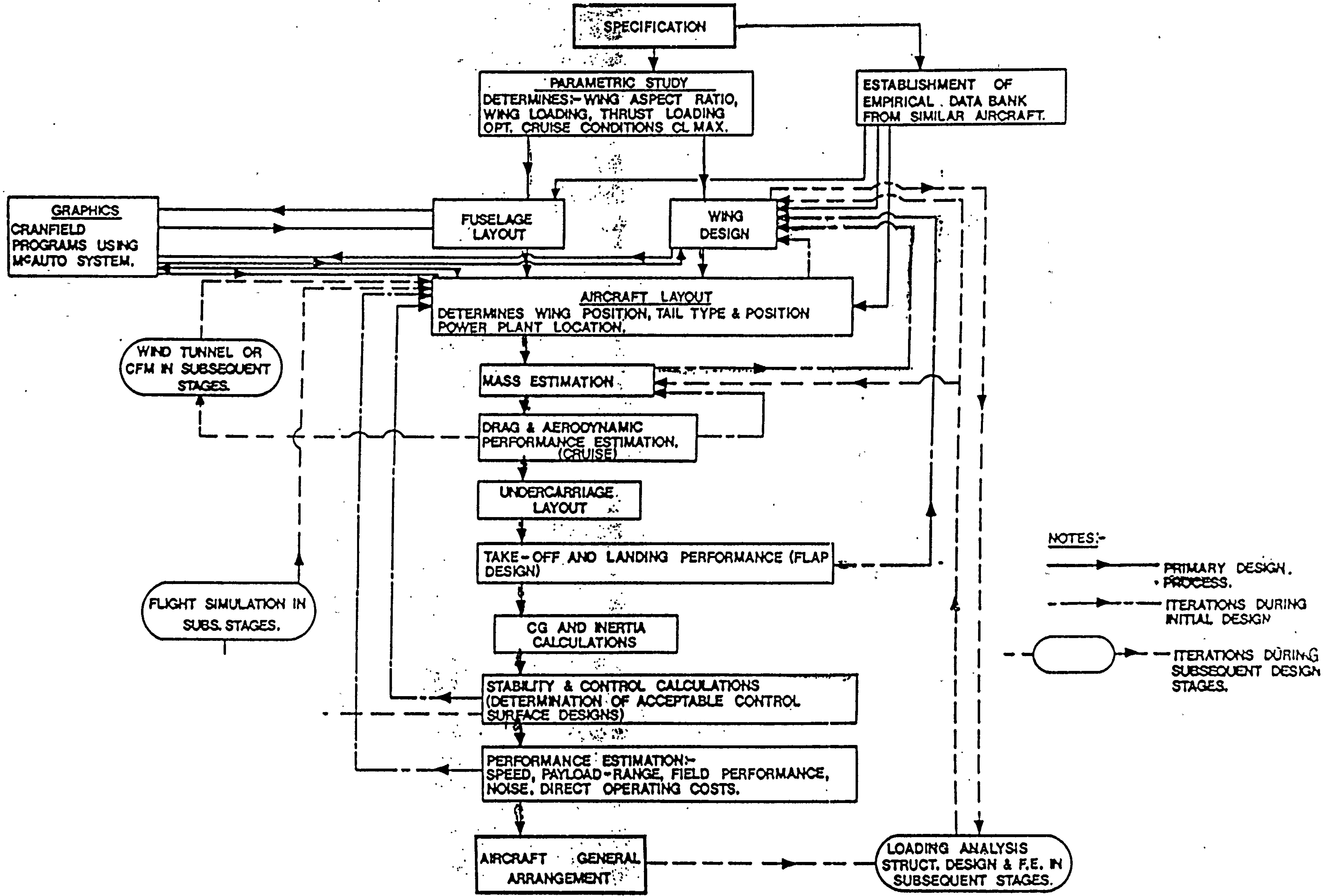


Figure 2.2/4 MAJOR STAGES IN AN AIRLINER INITIAL DESIGN PROCESS [2]

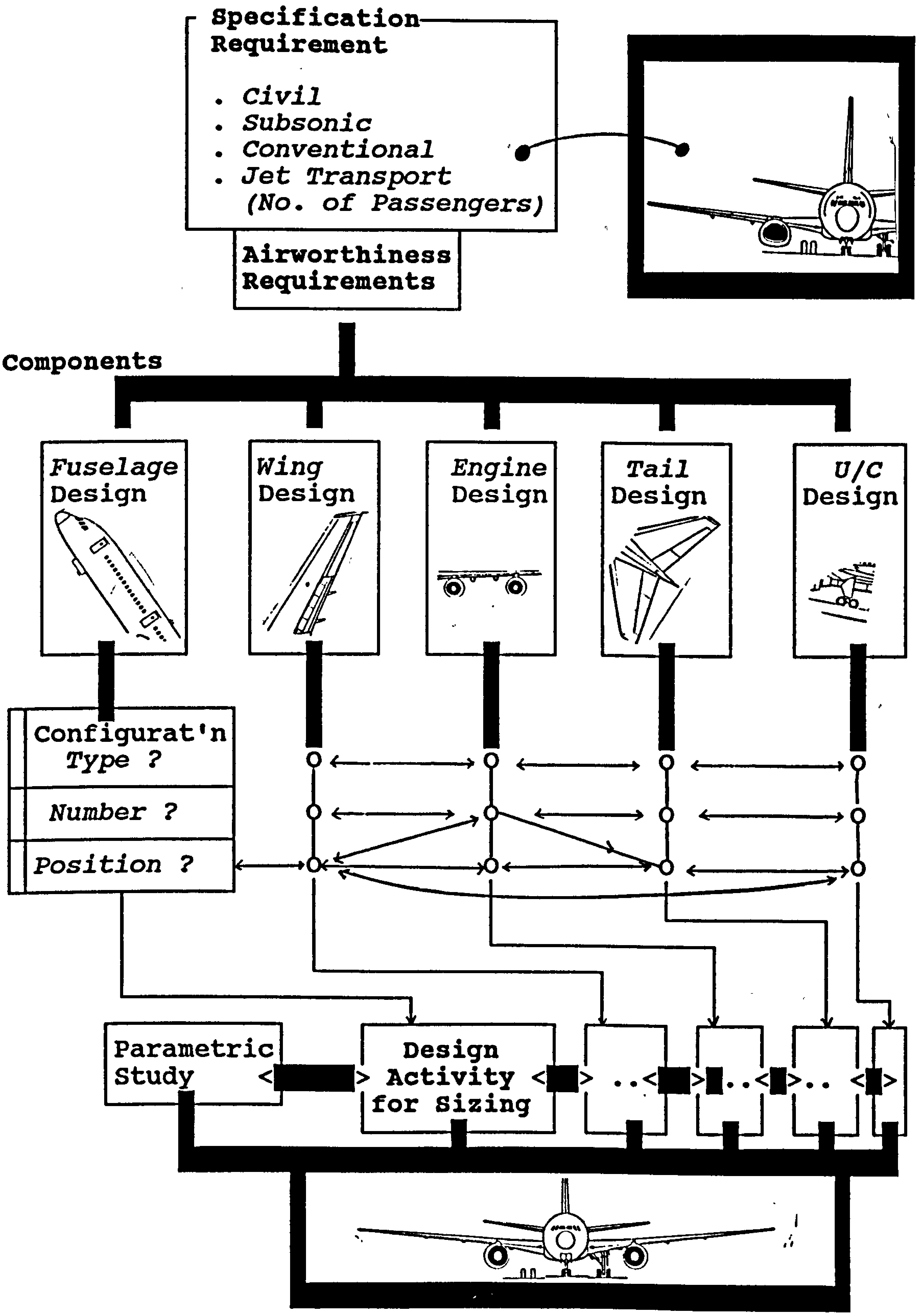


Figure 2.2.1/1 Aircraft Configuration Design

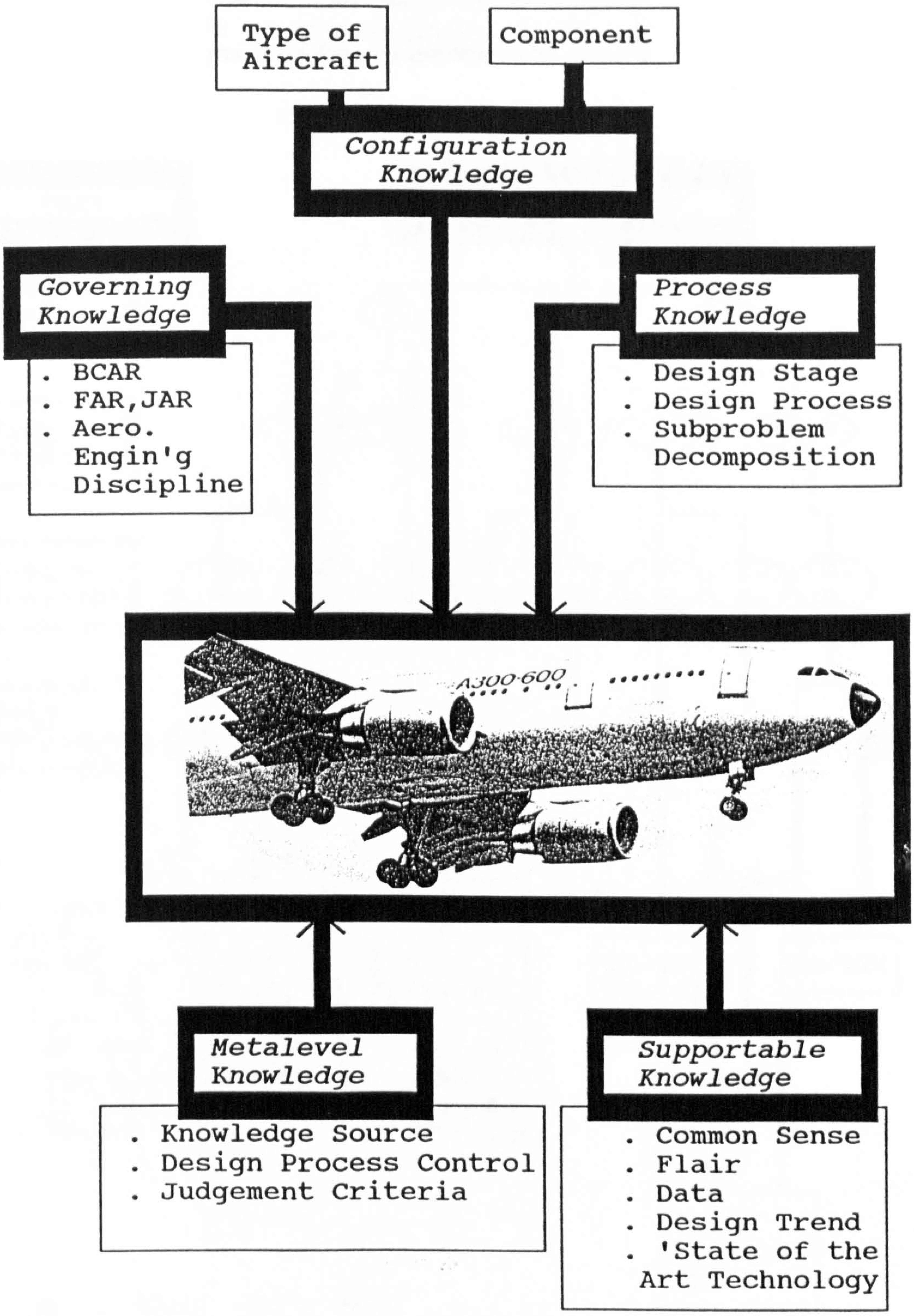
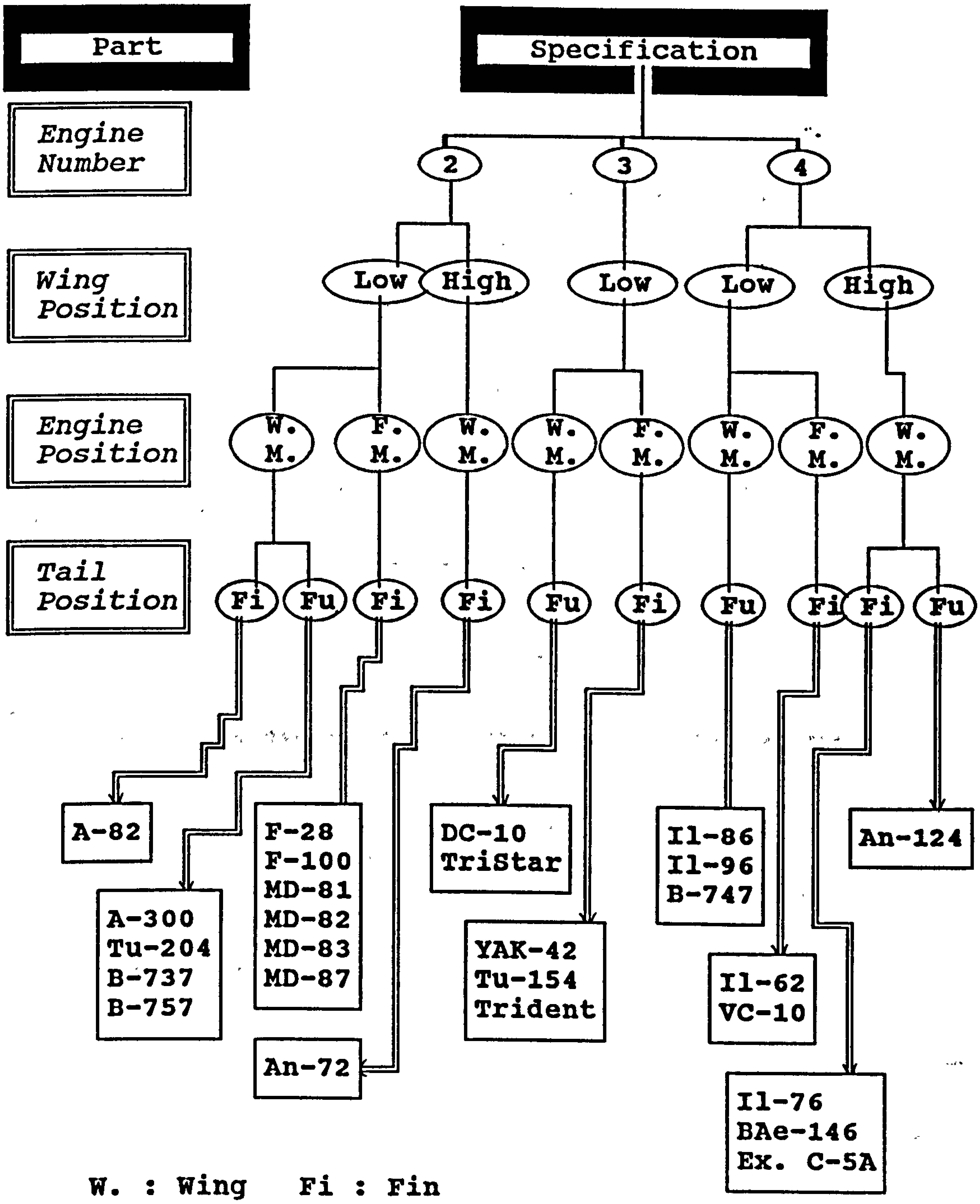


Figure 2.2.2/1 Aircraft Design Knowledge and Their Types

CONFIGURATION TREND



W. : Wing Fi : Fin
M. : Mount Fu : Rear Fuselage
F. : Fuselage

Figure 2.2.3/1 Configuration Trend

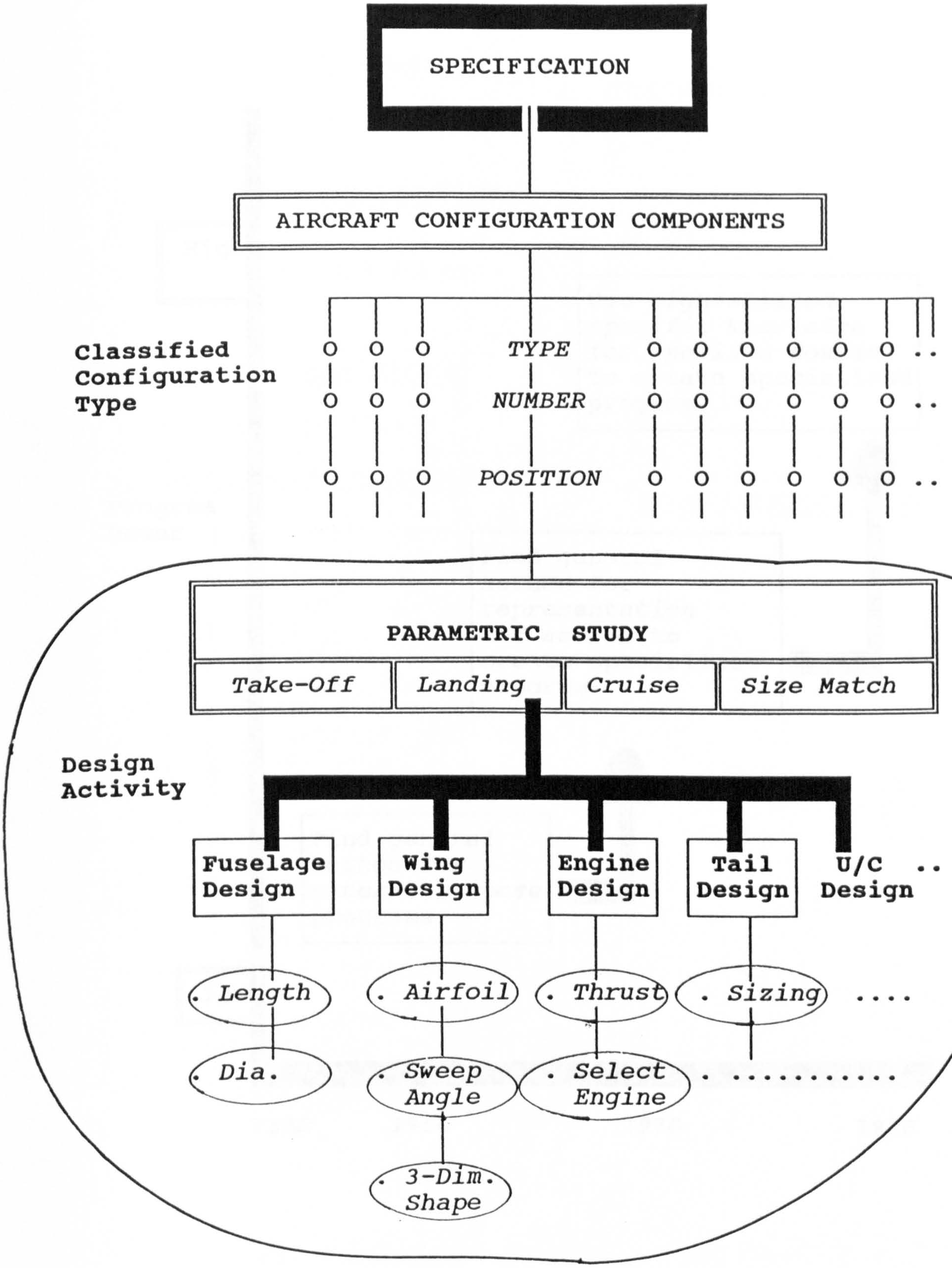


Figure 2.2.3/2 Design Activity for Aircraft Configuration Design

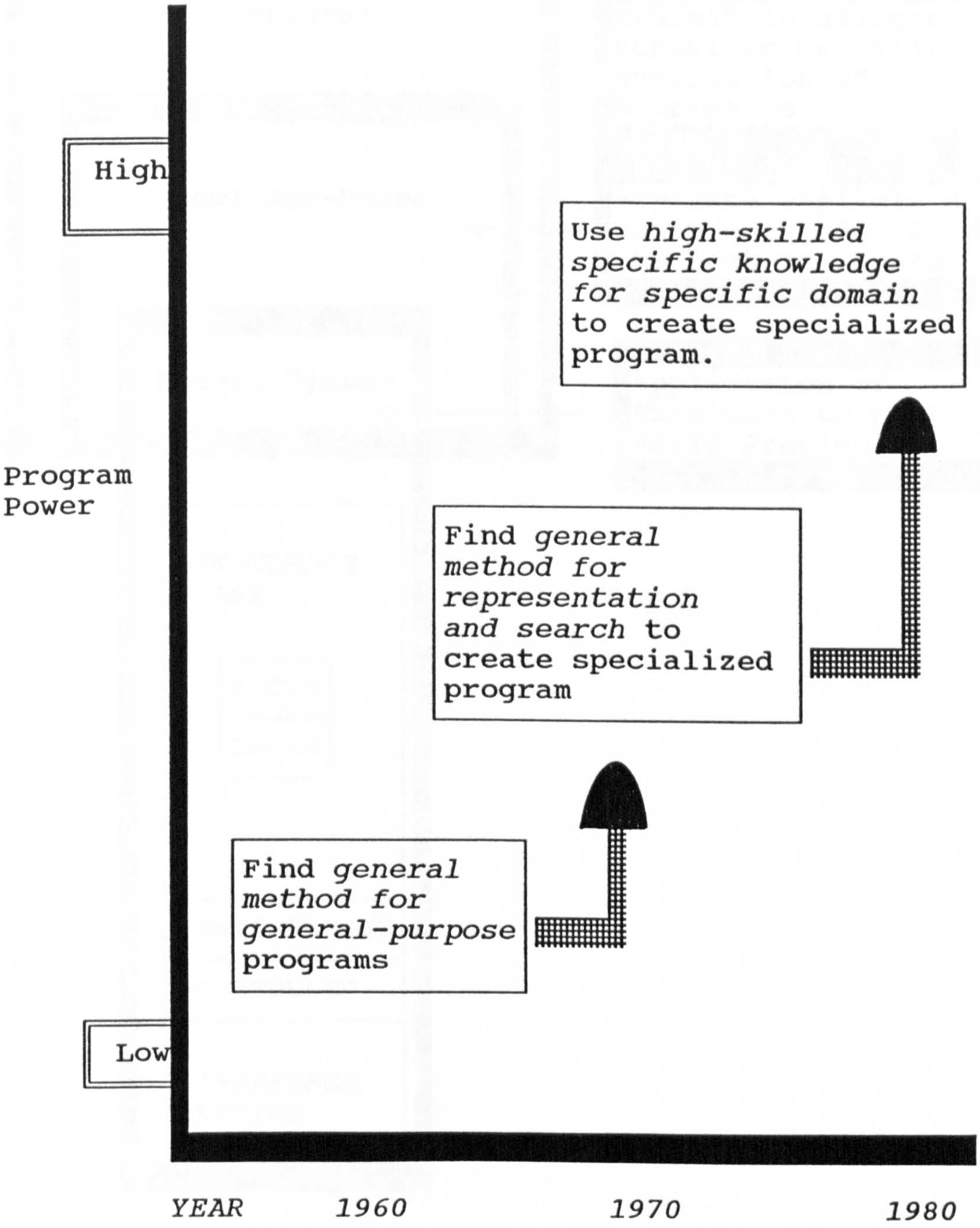


Figure 2.3.3.2/1 The Trend Shift in A.I. Research [61]

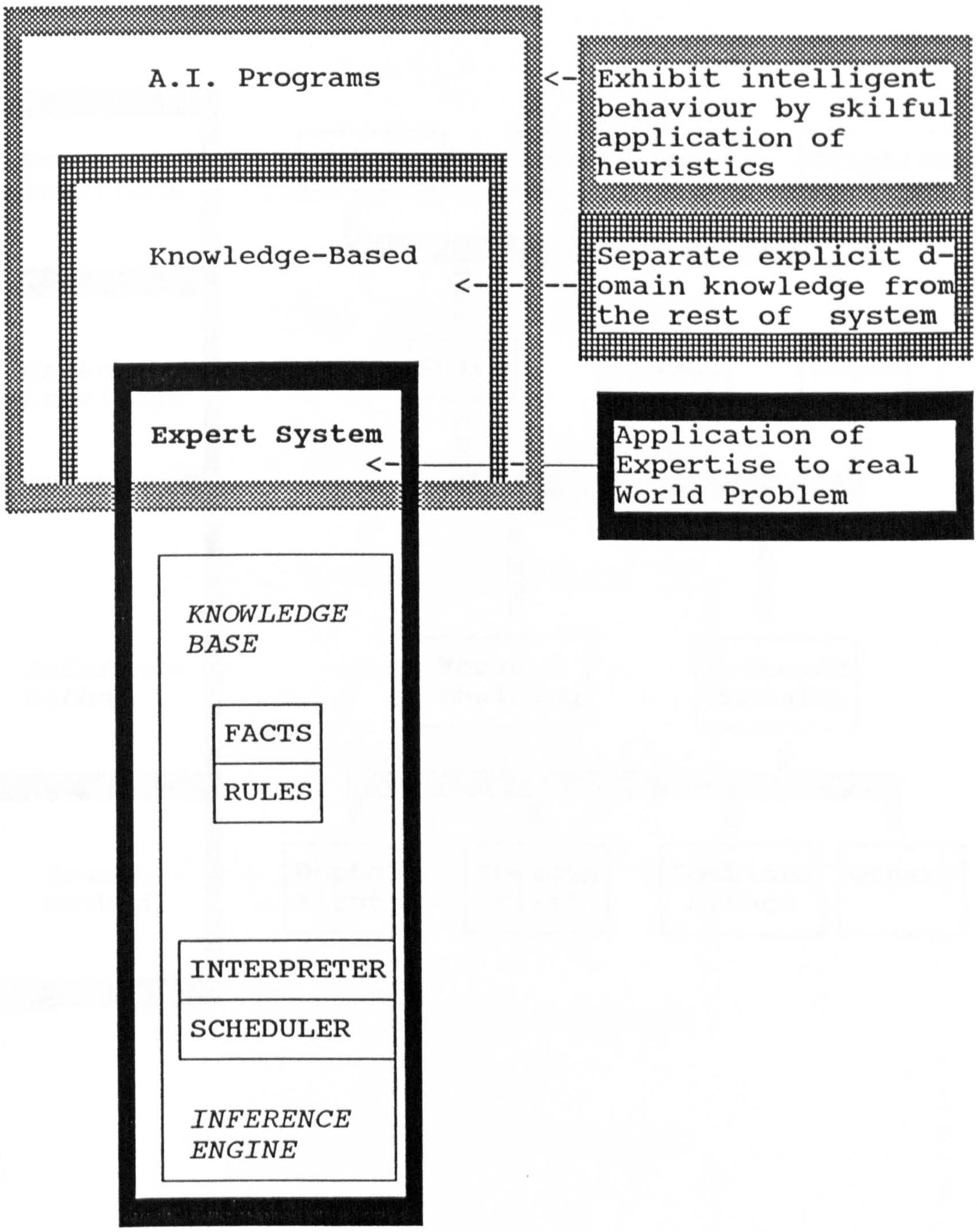


Figure 2.3.3.2/2 The Relations among A.I., Knowledge Based System, and Expert System [61]

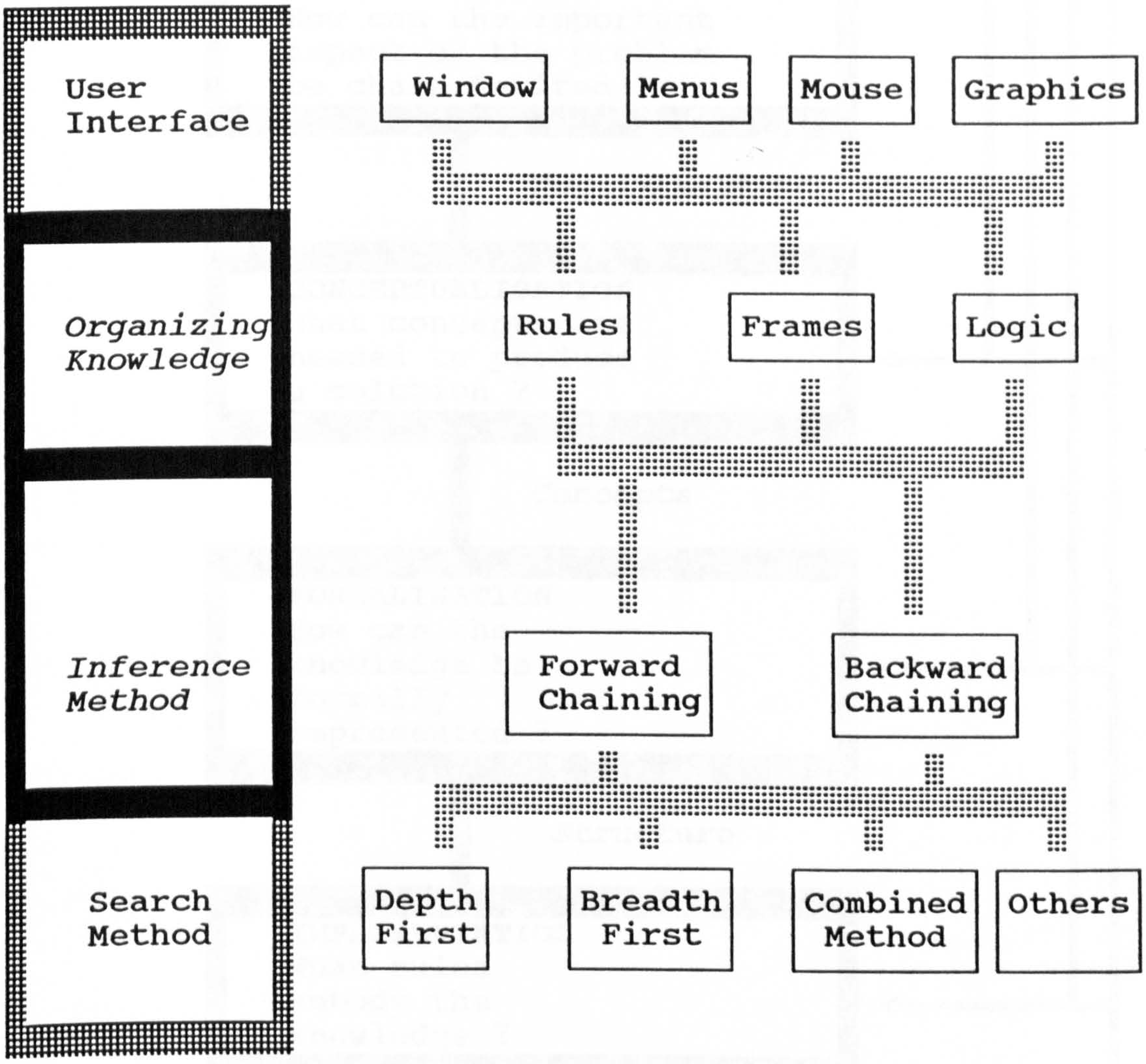


Figure 2.3.3.2/3 The Expert System Tool Structure [80]

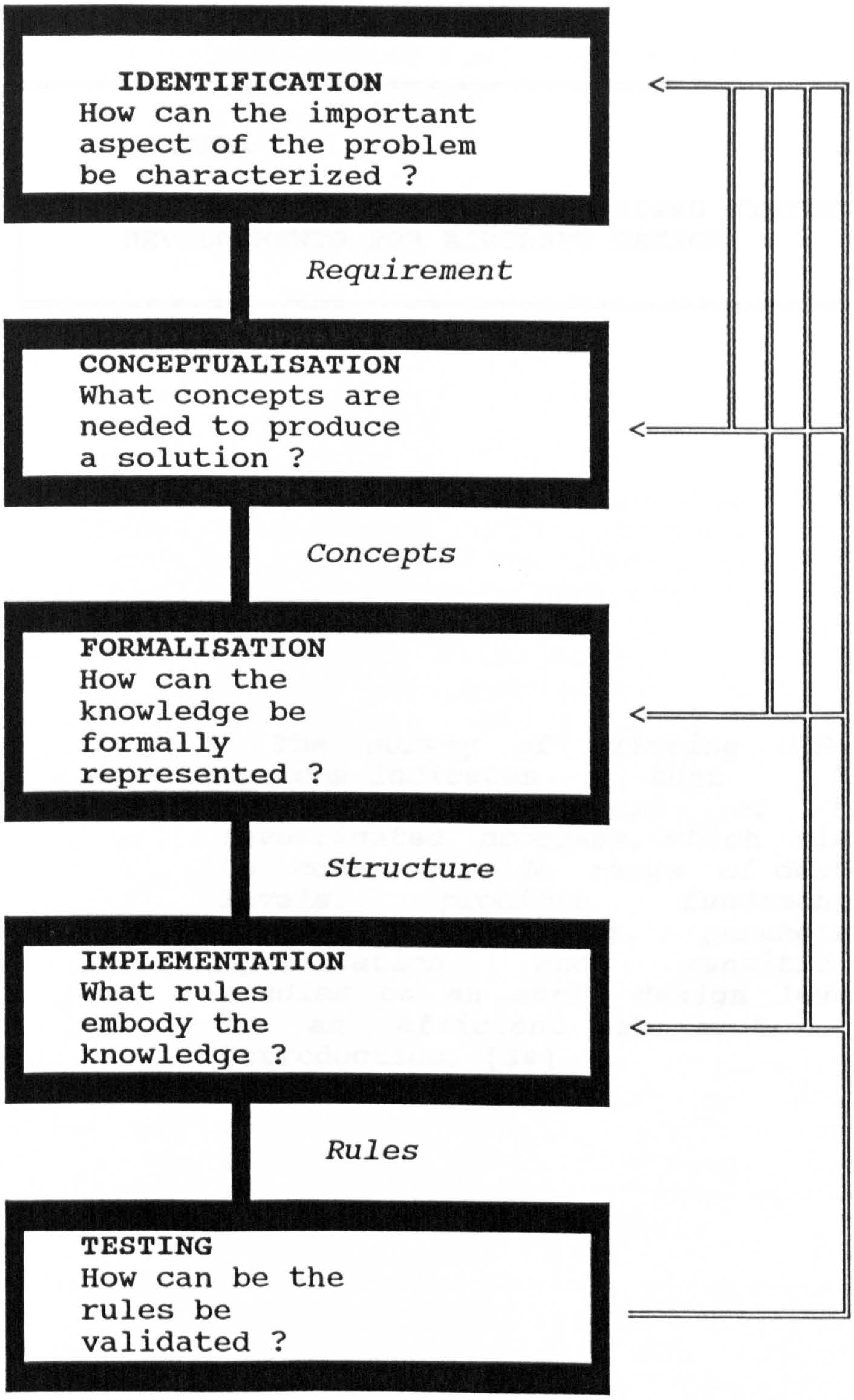


Figure 2.3.3.2/4 The Development Phases in Building Expert System [51, 52, 61, 79]

CHAPTER 3

LESSONS FROM PAST COMPUTERIZED SYSTEM DEVELOPMENTS FOR AIRCRAFT DESIGN

> *" The survey of existing CAD - Systems indicates that the complexity of most of the investigated programs, which claim to cover a wide range of design levels, prohibit fundamental investigations, e.g. parametric optimization and sensitivity studies on an early design level, in an efficient manner."* - Introduction, [34]

CHAPTER 3 LESSONS FROM PAST COMPUTERIZED SYSTEM DEVELOPMENTS FOR AIRCRAFT DESIGN

3.1 Introduction

As explained in the previous chapters and Appendix I, the aircraft design consists of complex processes which involve many interacting technical factors drawn from a wide spectrum of aircraft engineering disciplines.

Even so, there have been many attempts during the last two decades to overcome the complexities with the powerful computerized systems. Rapid progresses and brilliant advances in computer technology made it possible to apply a variety of computer techniques to an aircraft design, ranging from clumsy techniques at an early stage to efficient ones at present. [28, 29, 30, 31, 33, 34, 36, 38,39, 40]

Most of the computerized systems, in a strict sense, had limited applications in aircraft design disciplines. This is due to the fact that the scope of the engineering disciplines in aircraft design is too broad. They are also limited because the best configuration depends upon the importance the aircraft designer lays on design characteristics such as minimum weight, long endurance, long range, manoeuvrability, etc. Accordingly each computerized system has its own merit in some respects but unsatisfactory demerits in other respects.

Thus, the purpose of these sections is to review the computerized system used to support aircraft design and to extract those benefits transferable to other systems, which can lead to developing a better system for aircraft design in the future. Following this approach many features have been taken from the systems enumerated below and some of them are also incorporated in this Expert System.

1. SYNAC I and II (SYNthesis of AirCraft, [28])

2. *CPDS (Computerized Preliminary Design System, [29])*
3. *OPTIMIZATION [14] [30] [31] [33] [41]*
4. *GASP (General Aviation Synthesis Program, [32])*
5. *CAPDA (Computer Aided Preliminary Design of Aircraft, [34])*
6. *PASS (Program for Aircraft Synthesis Studies, [36])*
7. *ADROIT (Aircraft Design Regulation of Independent Task, [1])*
8. *First Step Toward Integrating the Design Process [37]*
9. *DSIDES (Decision Support In the Design of Engineering System, [38])*
10. *BIZJET [39] [40]*
11. *CASTOR [40] [41]*
12. *ACES [40] [42]*

3.2 Category of Computerized Systems

Four major categories of computerized systems were identified and consist of Procedural Numerical Analysis, Numerical Analysis plus CAD Package, Optimization Techniques, and the Expert System using the Artificial Intelligence.

1. *Procedural Numerical Analysis*

Among many programs developed so far SYNAC, GASP, BIZJET, and AAA fall within this category. [28] [32] [39] [43] These were used for the preliminary design stage and their particular feature is modularization, together with interactiveness and user-friendliness in case of the AAA.

Although these are useful tools for the aircraft design synthesis, these are specifically tailored to meet the requirements of a particular type of aircraft (BIZJET, GASP) in the preliminary design stage and thus they were not providing a variety of configuration types which must be considered at the conceptual phase.

However, the SYNAC tried to broaden its applicability through developing its derivatives and the AAA approach demonstrates the user-friendly utility by being equipped with a common data base built in the help files and report quality graphics for a configuration result. The general idea of SYNAC and its logic were well presented in the figure 3.2/1

2. *Numerical Analysis plus CAD Package*

As the computer techniques were progressed, a new field called the " Computer Aided Design " emerged. A number of pioneering efforts to incorporate a CAD methodology into the numerical analysis were made and the results were the CPDS [29], IPAD [44], CAPDA [34, STIDP [37], and AGPS [35]. Among these, the IPAD broadened its area to include spacecraft, but it is too voluminous to evaluate for this research.

An interdisciplinary technique, called the Computerized Preliminary Design System (CPDS,[29]), was computerized by the commercial airplane group of the Boeing Company among Aerodynamics, Configuration Design, Flight Controls, Propulsion, Structures and Weight Group.

This engineering tool made it possible to solve a wide range of the airplane synthesis problems and the tool aimed both at providing the maximum user acceptance by involving a familiar method and at minimizing the time to develop.

As shown in the Figure 3.2/2, an initial configuration is estimated in payload, range, size and speed. Then the system proceeds to the next step, the preliminary design, which performs the general arrangements how to locate wing, engine, fuselage interior layout, tails, and undercarriage, together with a noise treatment and the type of engines.

Then Design analysis checks that the results meet the performance objectives. Otherwise, cyclic iterations are implemented. And the off-design performances, manufacturing / operating cost, and marketability are evaluated. If the results are unsatisfactory, an iteration process starts. Otherwise, the detailed configuration and analysis are accomplished, which include a preliminary structural sizing, aeroelastic evaluation, flutter analysis, and elastic stability / control analysis. If they are satisfactory, a test is verified in the wind tunnel, structure and propulsion.

From the above, CPDS characterize

1. *Modular Programming* : Not a single program but a system of computerized methodology where the users can compile a computer program which was specifically tailored to solve particular design problems.

2. The maximum interaction between the user and computer.

3. An incorporation of the logical control mechanism

4. The CPDS uses 'well guessed' input data, which, in some sense, requires lots of experiences. Therefore only the design experts can use it and, moreover, it is well suited for the preliminary design.

The CAPDA program used an optimization as well and it is similar to the CPDS. However, it focuses on the parametric study and also it aims at showing the flexibility, transparency to the user of CAD system, user creativity, simplicity to allow the multivariate optimization, and expansibility to advanced design tasks.

The concept and structure of the CAPDA are shown in Figure 3.2/3. Other systems such as STIDP [37] for a configuration layout and AGPS [35] for the graphical surface geometry are used mainly for supporting the analysis through surface generation and they were well modularized.

3. *Optimization*

With the aid of a mathematical technique, numerical optimization techniques began to appear in the early 1970's and they are still receiving attentions. [30, 31, 33, 41, 14, 47]

There are many optimization programs available for a variety of applications, such as MVO [31], OPDOT [14], CASTOR [41], ADST [47], etc. The role of an optimization technique can speed up the process of converging the design variables within the constraints drawn from the design requirements. [40]

One of the main advantage lies in the wide selection of design variables and objective functions which can be optimized (e.g., cost and weight). However, the disadvantages are as follows;

1. Creating an mathematical model of the aircraft to be designed.
2. Formulating the design objective functions as a design variable, and selecting the constraints as a design variable.

As this technique starts from the mathematical model, the configuration phase is separated from the preliminary phase and thus it is limited in selecting /implementing many alternatives of the configuration.

4. *Expert Systems*

The systems reviewed in the above were limited in the concept application, considering the fact that the designer at an early stage of the configuration design should exercise his creative imagination on feasible configurations. In other words, they do not have the essentially-needed features such as the user's transparent full monitoring of what system is executing, the flexibility of altering the design sequence and its associated logic, and thus being able to examine diverse configurations. [51] [52]

Fortunately, the emergence of Artificial Intelligence computer programs gave a dramatic change to the development of aircraft design synthesis programs. Many systems appeared in the aircraft design area, examples of which are the PASS [36], DSIDES [38], ADROIT [1], ACES [42], etc.

Among these, ADROIT has some of the important expert system characteristics such as knowledge base, data bank, global controller for a reasoning process, interactiveness through user interface, and an explanation facility. Being initiated by ADROIT, the Artificial Intelligence group in the Cranfield Institute of Technology has implemented active research in the Expert System for the Aircraft Design, ranging from the configuration design, structural analysis, and CAD/CAM.

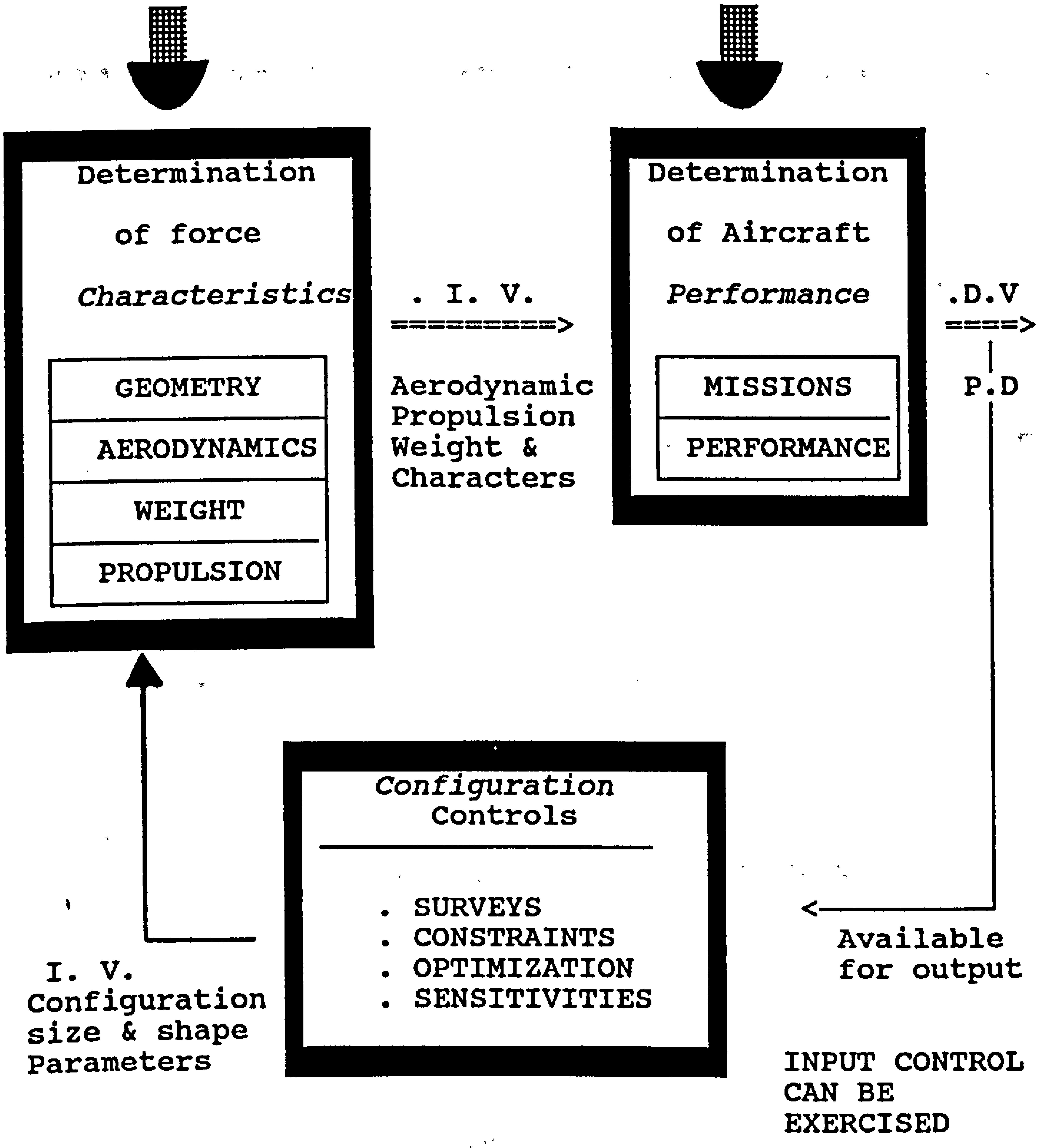
In case of the PASS, a solution can be found through a heuristic search and the DSIDES adopted the optimization - like features together with the A. I. techniques, to support the designer's judgement. The main features of these Expert Systems incorporate a high degree of the modularity, not only to encourage extensibility but also to reduce complexity.

3.3 Comments on the Review

As shown in the summary of the Figure 3.2/4, some of the commonly - used features have high modularity and interactiveness. In the expert system, was added the control mechanism for reasoning to find the path for a solution and explanation facility.

The main drawbacks of most existing expert systems are the limited scope in the aircraft category, the lack of a full explanation facility for reasoning processes, and the want of creative imagination at the configuration phase.

In conclusion, the author tried, first, to widen the aircraft category from that used in other systems, second, to incorporate an efficient control mechanism, third, to give a full explanation of the clear reasoning processes, and finally to explore various configurations. Also the author believes that these efforts enhance the expert system's capability for handling aircraft design with respect to such features as the transparency of complex design processes, extensibility of control mechanism to other system, flexibility of changing design knowledge base, and incorporation of user creativity. [34] [36] [40]



D. V. ; Dependent Variables
P. D. ; Performance Data
I. V. ; Independent Variables

Figure 3.2/1 The SYNAC's General Idea and Logic [28]

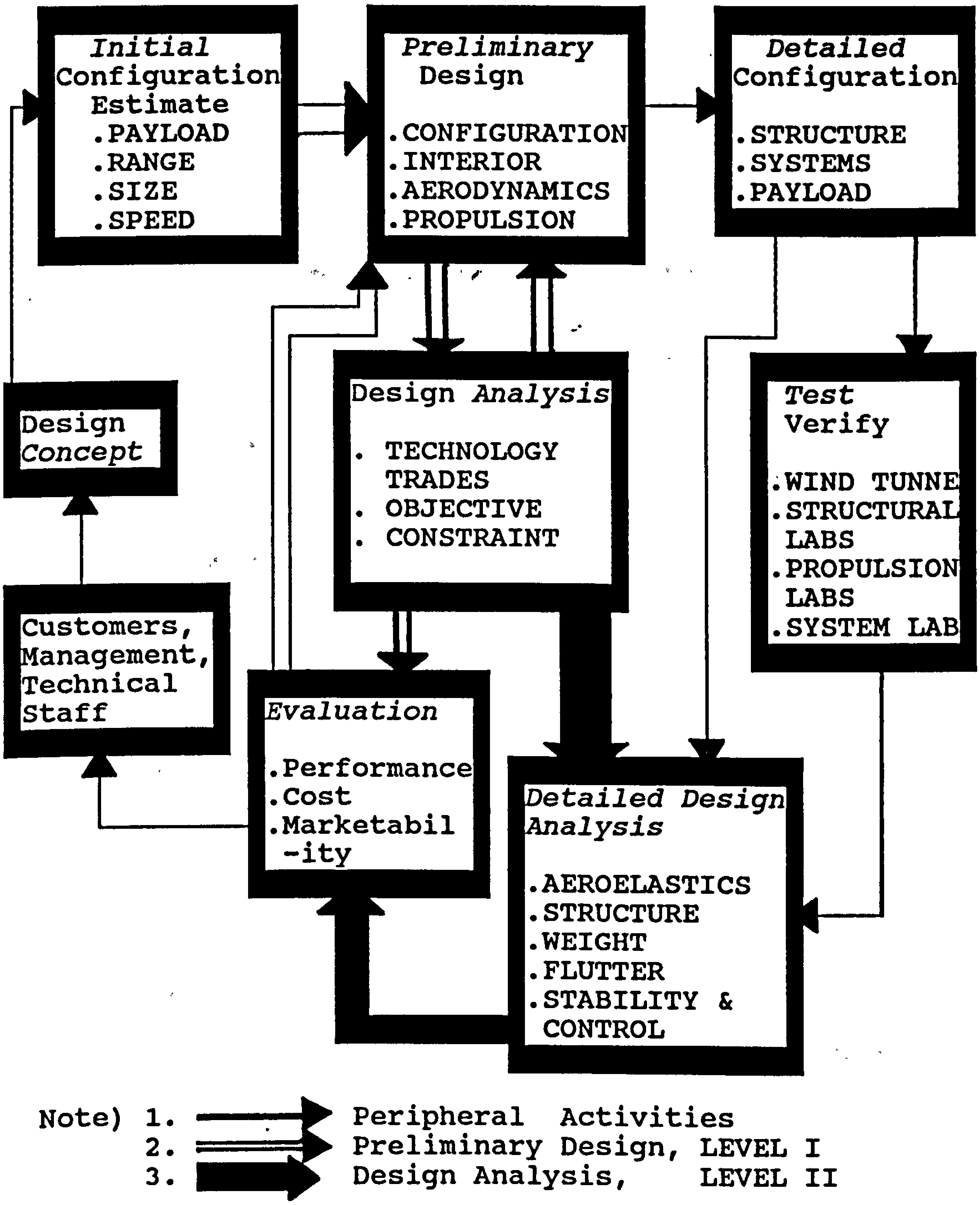


Figure 3.2/2 The CPDS - Preliminary Design Flow Chart [29]

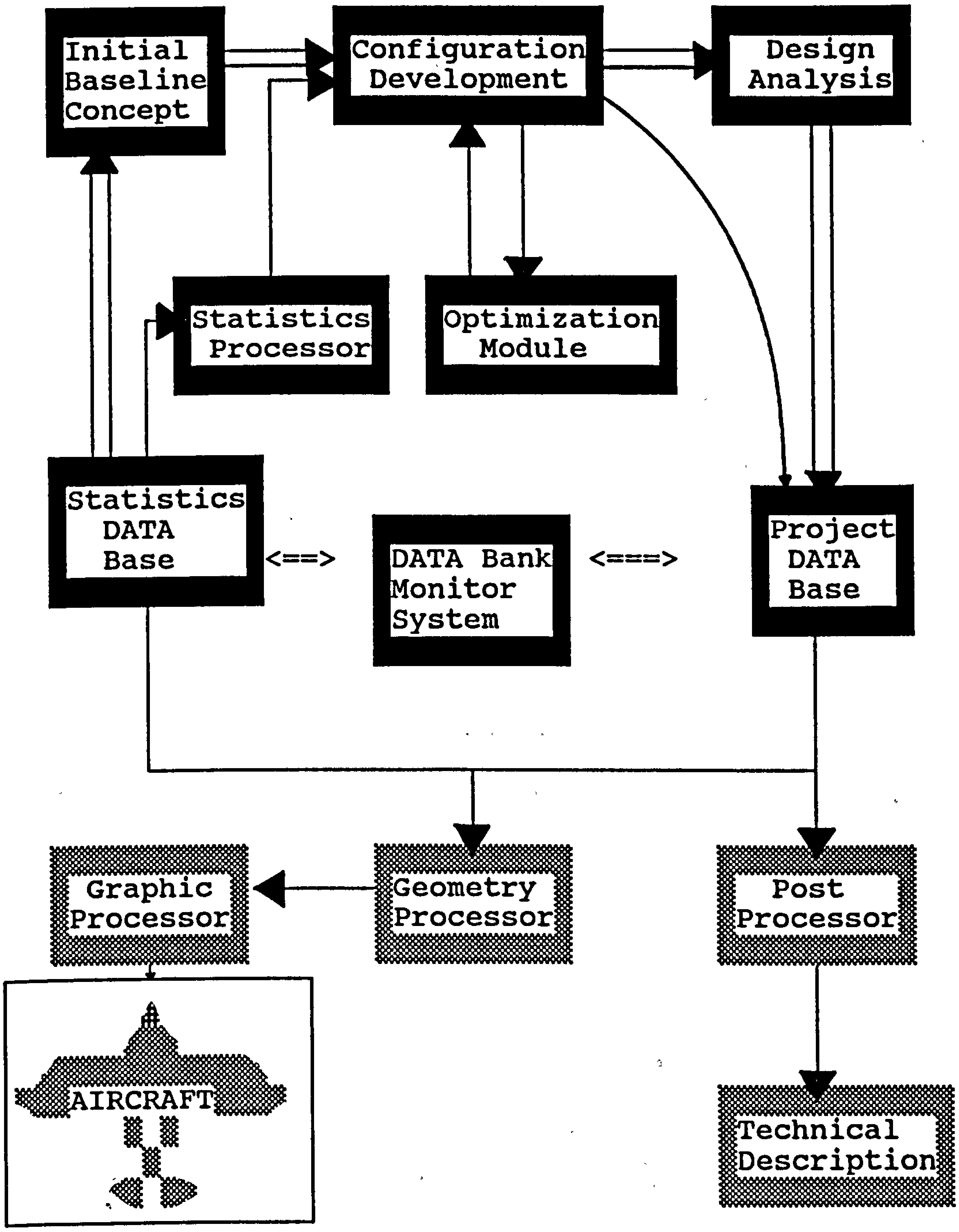


Figure 3.2/3 The Concept and Structure of CAPDA [34]

CATEGORY	SYSTEM	SPEED	PHASE	FEATURE
Procedual Analysis	SYNAC[28]	sub/sup	preli.	Modular
	GASP [32]	sub	"	Mod
	BIZJET[39]	sub	"	Mod
	AAA [43]	sub/sup	"	User Friendly
Procedu. + CAD	CPDS [29]	sub/tra	"	Mod/Int
	CAPDA[34]	sub	"	Mod/Opt
	STIDP[37]		"	Geo/Sys
	AGPS [35]		"	Sur/Gen
Optimization	OPDOT[14]	sub	"	Con/Par
	NumOpt[30]	sub	"	
	MVO [31]	sub	"	Mul/Opt
	ConOpt[33]	sub	"	Con/Opt
	CASTOR[41]	sub	"	
	ADST [47]	sub/hyp	"	Modular
Expert System	PASS [36]	sub	"	Mod/K.B
	DSIDES[38]	sub	"	Int
	ADROIT[1]	sub	"	Reason'g
	ACES [42]	sup	"	Reason'g

Note) 1. Mod. : Modularization
2. sub : Subsonic
3. sup : Supersonic
4. Pre. : Preliminary Design
5. Int : Interactive
6. Opt : Optimization
7. Geo/Sys : Geometry System
8. Sur/Gen : Surface Generation
9. Con/Par : Constrained Parameter
10. Mul/Opt : Multivariate Optimization
11. K.B : Knowledge Base
12. Reason'g: Reasoning and Inference

Figure 3.2/4 The Summary of Review on the Aircraft Design Systems

CHAPTER 4

ANALYSIS OF AIRCRAFT DESIGN KNOWLEDGE

" *There is a good rule in design; ..... K I S S system, Keep It Simple and Stupid!*" - Introduction, [22] [2]

CHAPTER 4 ANALYSIS OF AIRCRAFT DESIGN KNOWLEDGE

4.1 Introduction

Unquestionably, aircraft design procedure starts by speculating on the requirements stipulated in the specification. The next step is generally called the "Conceptual Design" where a designer conceives the general arrangement / layout for configuration components. Another step called the " Preliminary Design " follows to quantify their numerical dimensions and to analyze their required characteristics. The last step called the " detail design " should be implemented for manufacturing but was excluded here.

However, the design procedure can vary according to the designer's intention and interest. Many kinds of design procedure could be found. [2] [3] [6] The author modified the procedure of Figure 2.2/4 "Cranfield Method" for the application to this Expert System and this consists of the following three phases.

1. The **Set-Up Phase** for speculating a specification and general requirements.(*)

2. The **Configuration Phase** for the general Arrangements of components and their integrated layout.(*)

3. The **Design Phase** for sizing each component and analyzing its associated design characteristics.

4.2 Set-Up Phase : Specification Requirements

(*) These phases can be regarded as a conceptual design, sometimes including the parametric study of Design Phase.

A specification includes any combination of the performance objectives such as the purpose of use, number of passenger, cruising speed, range, rate of climb, cruising altitude, type of engines, etc. They influence design of each component. For example, the number of passenger is closely coupled with Fuselage Diameter, Fuselage Length, Payload, and Total Weight. Therefore, they were analysized in the 'Specification Study' of Appendix I, together the following considerations.

- *Type : Aircraft*

- *Purpose of Use : Civil or Military*

- *Category of Aircraft : Transport or Fighter*

- *Speed Range: Subsonic, Transonic, Supersonic, Hypersonic.*

  1. Subsonic ; if the Cruising Mach Number M_{cr} is less than 0.99.
  2. Transonic ; if the Cruising Mach Number M_{cr} is greater than 0.99 and less than 1.2.
  3. Supersonic ; if the Cruising Mach Number M_{cr} is greater than 1.2 and less than 4.0.
  4. Hypersonic ; if the Cruising Mach Number M_{cr} is greater than 4.0.

- *Configuration Concept : conventional or unconventional*

- *Take-off and Landing Concept :*

  1. CTOL : If the Take-off field length is greater than 5000' and the Landing Field Length is less than 7000'.
  2. STOL : If the Take-off field length is less than 5000' and the Landing Field Length is less than 5000'.
  3. VTOL : If the Take-off field length is equal to 0 and the Landing Field Length is equal to 0.
  4. STOVL: If the Take-off field length is less 5000' and the Landing Field Length is equal to 0.

4.3 Configuration Phase : Configuration Components

The required components must be selected properly for a integrated configuration and the designer should decide the following type, number, position of each component, thus making general shape of aircraft.

- *Component : Fuselage, Wing, Engine, Horizontal Tail, Vertical Tail, and Undercarriage.*
- *Component's Type : Every Type possible.*
- *Component's Number : Such numbers as 1, 2, 3, 4, and 5, as are practicable.*
- *Component's Position : Every Position that can be realistic.*

Their classified description is well summarized in Figure 4.3/1. As shown in Figure 2.2.3/1 "Configuration Trend", the position of the wing and engines influences the horizontal position, resulting in different configurations. Their pros and cons are described in Figures 4.3/2 and 4.3/3.

4.4 Design Phase : Preliminary Design Activity

4.4.1 Introduction

In this Phase, the designer performs various kinds of design activity such as the parametric study, wing design, fuselage design, engine design(selection), horizontal and vertical tail, undercarriage design, weight / cost analysis, etc. The required items for analysis are well summarized in Figure 4.4.1/1 and the analysis is implemented in the following order.

1. Design Consideration for each component design.
2. Detailed Analysis from engineering disciplines.
3. The Input / output data.
4. The process and its related equations and calculations.

All the items of the above No. 2, 3, and 4 were described in Appendix I "Detailed Analysis of Aircraft Design Knowledge".

4.4.2 Parametric Study

4.4.2.1 Specification Study

To begin with, the following requirements for Modern Jet Transport need to be scrutinized.

1. *The Number of Passenger*
2. *Flight Crew and Cabin Staff*
3. *Flight Range*
4. *Maximum Payload*
5. *Flight Altitude*
6. *Cruise Speed (Cruise Mach Number)*
7. *Take-Off / Landing Field Length (F.A.R. Requirements)*

The analysis of these requirements is in Appendix I.

4.4.2.2 Parametric Study

The parametric study(*) is very important in that it estimates parameters which form the basis of dimensioning the parameters in the wing, fuselage, and overall configuration components, as shown in 4.4.2.2/1.

According to the FAR part 25 for all the commercial Jet Powered Transports, the designer must meet at least the following performance objectives.

1. Airport Performance :
    1. FAR Landing Field Length
    2. Missed Approach requirement
    3. FAR Balanced Take-off Field Length.
    4. Second Segment Climb Gradient
2. Cruise Performance :
    1. Cruising Mach Number
    2. Cruise Range
    3. Payload

It is the purpose of this parametric study to find a realistic wing loading and its related thrust loading, which are matched among the following requirements.

1. *LANDING PERFORMANCE AND MISSED APPROACH*
2. *TAKE-OFF AND SECOND SEGMENT CLIMB*
3. *CRUISING PERFORMANCE*

In finding the matched points shown in Figure 4.4.2.2/2, the coordinates of wing loading (X axis) and thrust

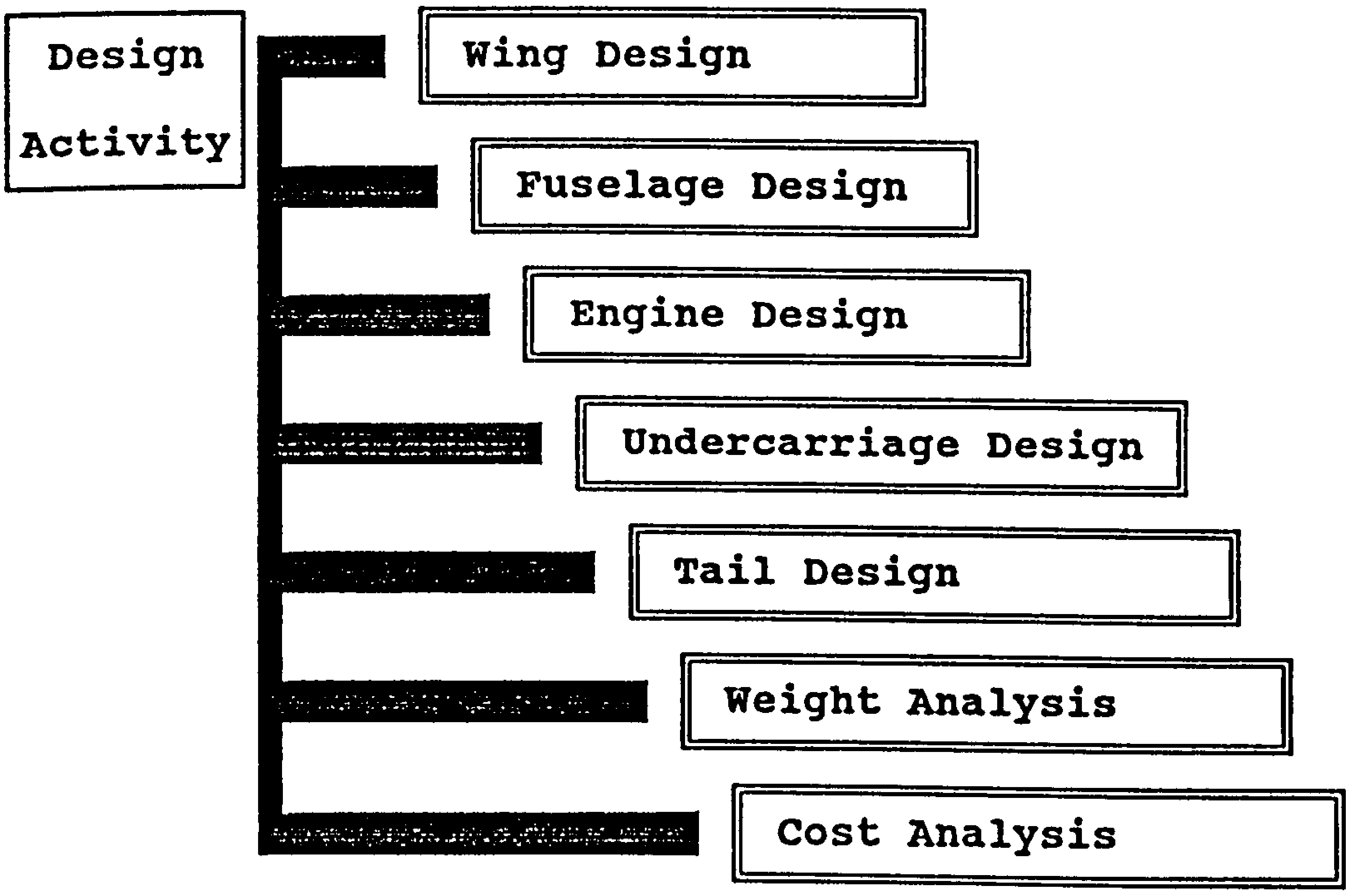
---

(*) The analysis with respect to the parametric study was based upon the NASA report [7].

loading (Y axis) obtained from the performance calculations per landing, take-off, and cruise condition are connected with lines. The intersection points among the lines are the matched points and these matched points can decide the corresponding parameters relevant to parametric study, as shown in the parametric study of Appendix I. These resulting parameters, which are achieved from the matched points pertinent to all the above conditions and which are also required for design activities, are as follows;

1. GROSS WEIGHT (W_G)
2. EMPTY WEIGHT (W_e)
3. FUEL WEIGHT (W_f)
4. WING AREA (S_W)
5. TOTAL THRUST (T_O)
6. FUEL CONSUMPTION (s.f.c)
7. FUSELAGE DIAMETER (Dia or d)
8. FUSELAGE LENGTH (L_F or l)

All these parameters are closely related with the following design activities, as explained in Appendix I.



4.4.3 Wing Design Analysis

4.4.3.1 Design Consideration

Once a designer defines the role of aircraft and its speed range, the design of the wing is initiated and it must be carefully accomplished because it critically affects the range, maximum speed, manoeuvrability, etc. Thus, the designer must carefully consider the aspects which affect the wing design such as the aerodynamics and structure.

The designer defines, from an aerodynamic point of view, the *wing planform, thickness ratio, taper ratio, twist ratio, incidence angle, swept angle, etc.* From the structural point of view, he arranges the wing rib, spar, material to be used, etc., to sufficiently withstand the loads generated during flight. This structural design is closely related with a wing weight. In this regard, the following considerations should be given to wing design. [2] [3] [4]

1. Aerodynamic consideration

As it has already been decided in the configuration phase *where to position a wing among the low, mid, or high locations* on the centre fuselage, the effects of wing location will not be repeated here. Instead they are well summarized in the Figure 4.3/2. Thus, the main aspects to be considered at this stage are as follows;

1. The main role of a wing is to generate a lift. The lift is greatly affected by a wing section and shape under the control of designer, wing area, angle of attack and speed under the control of pilot and air density.

2. The Drag, which is composed of the zero lift drag and lift dependent drag, must be minimized. As the speed increases near a sonic range, the onset of effects of compressibility (or drag rise) appears and methods to delay them are necessary. The methods include sweep-back(*), low wing loading, low aspect ratio, and supercritical sections(**).

3. The Flying qualities such as stall, buffet, and stability problems are particularly influenced by a wing design. The need to provide a high speed aircraft with good low speed flight characteristics often gives rise to a conflicting situation.

   _. In equation $V_{stall}^2 = (W_G/S) / ((1/2)*Rho*C_L)$, the stall speed is determined by the wing

---

(*) Swept-forward wing was not considered here but it was referred to in the section of Chapter 2, the "State of the Art" Technology.

(**) These airfoils of RAE 95 series were incorporated into this expert system.

loading and Maximum lift coefficient whilst the stall behaviour is governed by the wing planform, airfoil section, and wing twist.

- Due to high speed manoeuvres or gusty whether, the aircraft often experiences buffet phenomena.

- At the high subsonic speed, the danger must be alleviated of suffering from the longitudinal instability (tip - stall, tuck-under, and speed instability), lateral - directional stability (poor dutch roll damping, wing drop or wing rocking), and lateral control deficiencies (aeroelastic deformation at high EAS, aircraft dynamics at high lift).

2. Structural Consideration

1. The Aeroelastic effects such as flutter (torsional, flexural) and aileron reversal must be reduced so as not to break-off the wing. Thus the wing structure must have adequate strength and stiffness in bending and torsion.

2. The Long-life characteristics can be achieved with good fatigue / creep properties by a correct choice of materials and the reduction of stress levels, and with good corrosion properties by surface finish / protective treatment / electrolytic action.

3. For a successful aircraft, it is essential to achieve a Minimum weight in the wing structural design. Of course, the detail design influences the weight. However, as one way of reducing weight, the use of composite material is today's trend and thus its usage in the wing structure must be estimated.

4. Additional attentions must be given to sufficient fuel volume capacity, the aerodynamic control devices such as flaps / aileron / spoilers / slats / airbrakes, leading edge device, engine position (under_wing mounted or rear_fuselage mounted), and undercarriage.

Summarizing the above characteristics required for the preliminary purpose, the designer must understand the effects of airfoils, sweep-back angle, twist angle, aspect ratio, thickness, dihedral (or anhedral) angle, incidence angle, ratio of composite material used, "state of the art" technology such as the active control technology, etc. All the effects are well expressed in the Figure 4.4.3.1/1. Then, the next procedure is not only to select an appropriate two dimensional airfoil but also to measure the three dimensional drag_rise Mach No., thickness ratio, aeroelastic check, tip stall, flap, and finally wing weight.

In consideration of the above requirements, it is by no means an 'easy-going' discipline to size and determine a wing shape. Fortunately, the Cranfield Institute of Technology initiated an application of Artificial Intelligence Techniques to aircraft design a few years ago and the first result was the wing design expert system called *"ADROIT(Aircraft Design by Regulation of Independent Task)"*. [1]

However, a review of ADROIT found the fact that factors such as Wing Taper Ratio, Dihedral Angle, Incidence Angle, and Twist Angle were not calculated. Instead, their values based upon designer's experience must be inserted.

This wing design expert system was found to be well suited for the initial design stage and the author tried to incorporate it into this Expert System with revisions.

4.4.3.2 Wing Design Process

With the above considerations, the designer must consider the following effects, firstly with respect to the 2 dimensional airfoil characteristics, and secondly with respect to the 3 dimensional effects.

1. The Two dimensional airfoil characteristics include such phenomena as

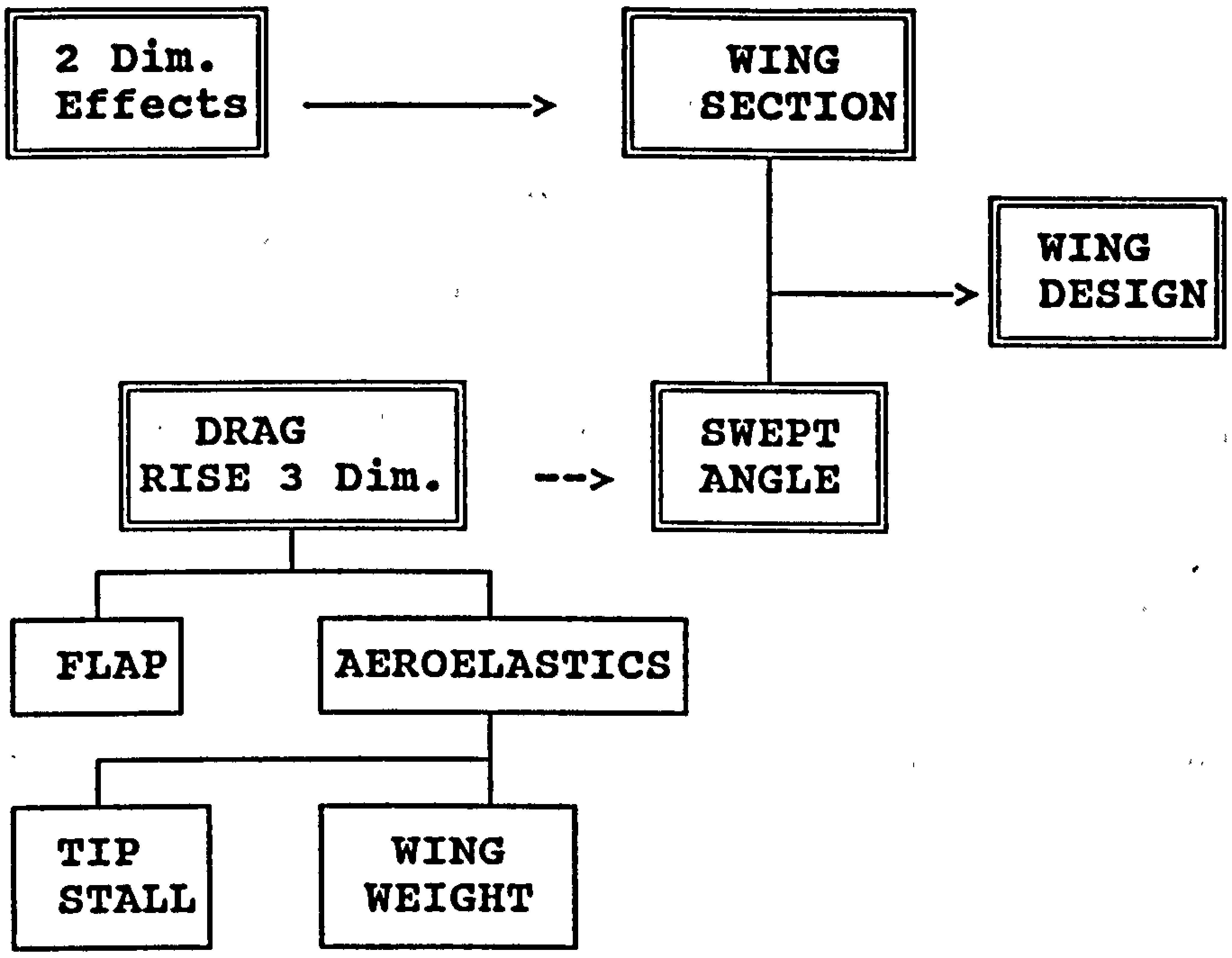
    1. Low speed lift coefficient
    2. Stall characteristics
    3. Cruise Lift/Drag Ratio
    4. Pitching moment
    5. 2-dimensional drag-rise.

2. The Three dimensional effects must consider such phenomena as

    1. 3 dimensional drag-rise Mach No.
    2. Thickness ratio
    3. Aeroelastic check
    4. Tip stall check
    5. Flap effectiveness
    6. Wing weight estimate

    These effects have particular influences on estimating the thickness ratio, aspect ratio, swept angle, and wing weight.

Thus, the overall procedures can be briefly described in the following diagram.



4.4.4 Fuselage Design

First of all, the fuselage must give structural integrity and useful volume to an aircraft. Preliminary general arrangements for aircraft components are closely linked with a fuselage design. In fact, although the effects from locating the other components must be accounted for and accordingly the variations associated with the effects are inevitable, the fuselage design is commenced as a natural starting point before sizing the other components and the overall configuration. Thus the main characteristics tied up with its design can be described as follows; [2] [3] [4] [5]

1. From a conventional configuration point of view, it is most efficient to consider two important clustering effects. First, the location cluster, *i.e., what is housed where - the aft section contains the APU, the tail surfaces etc.* Second, the associated clusters which explain *what is joined to what, i.e., wing to fuselage, engines to wing, etc.* In addition, the fuel

storage in the fuselage must be considered to maximize the overall fuel volume, even though undesirable.

2. The type of fuselage decided on is between the circular, double - bubble. Its drag, representing 20 to 40 % of the zero lift drag, must be sufficiently low and thus the cross section width or diameter must be also minimized, with the overall length optimized.

3. During the Landing, the Pilot in Command (seated usually on the left side of cockpit) must have an angle of view 15°, downward without having to rely on an undue head movement.

4. The jet engine installations should be carefully treated so that the jet exhaust and noise cannot damage the structure.

5. The volume allowed for stowing the undercarriage must be also reduced as compactly as possible.

6. In civil airliners, it should be borne in mind that *the fuselage volume is decided by number of passengers and freight capacity*. Also, for passenger comfort, the flexibility of the layout should be maximized.

7. From a structural point of view, the basic philosophy of a structural design is to obtain a high level of safety with adequate residual strength, ease of maintenance and inspection, and the minimum structural weight. Especially as an aircraft may accumulate up to 60,000 flying hours during the 15 more or less year periods, without major rebuild this leads to serious design problems such as the fatigue and corrosion.

8. The *amount of composite material* used must be taken into account.

Due to limitations in the memory capacity of the available PC used in this study, the designs of the cockpit, internal arrangement, nose fuselage, and tailcone are not incorporated. Instead, the fuselage length, diameter, and weight are checked with respect to an economy class transport. Of course, the designer can mix the first, normal, and economy class.

4.4.5 Engine Design (Selection)

The engine design can be regarded as an independent design area and therefore it is extremely difficult to implement the engine design completely. Most of aircraft designs incorporate existing engines suitable for the desired aircraft performance. However, if a new engine must be used for the aircraft to be designed, then the new engine is generally designed in line with the progress of new aircraft development.

The designer must bear in mind that the types of engines most suitable for the performance requirements are governed by the following characteristics.

1. The normal flight speeds during operation are decisive in selecting a proper type. For example,

Mach Number Range	Suitable Type of Engine
$0 < M_{cr} <= 0.5$	Piston Engine
$0.5 < M_{cr} <= 0.6$	Turboprop Engine
$0.6 < M_{cr} <= 0.99$	Propfan or Turbofan
$1.0 < M_{cr}$	Turbofan or Turbojet Engine

2. The Hourly Fuel Consumption should be as low as possible.

   . Thrust Specific Fuel Consumption (C_T) for Jet Engine

   . Specific Fuel Consumption (C_p) for Propeller Engine compared with $C_p * V$ / (propeller efficiency).

3. The Installed Engine Weight, which is largely dependent upon the performance requirements, should be as light as possible, because the power plant weighs more or less 20 % of the aircraft empty weight.

4. The means must be sought to suppress the engine noise (or external noise such as fan / propeller noise). In particular, the jet exhaust noise is the cause of both passenger discomfort and actual structural damage.

5. As for safety aspects, the power plants must be mounted away from the passenger compartments with a fuel line / tank kept far away from the hot parts of the engine. In particular, the propeller should be kept out of line with passenger seating.

6. Adaptability, which is the ability to change a type of power plant with a little structural revision, shall be borne in mind due to today's rapid progress of the

engine technology. In this regard, the podded engine appeals more than engines mounted elsewhere.

7. The Auxiliary Power Unit (APU), which is a small gas turbine equipped into almost all the transport, supplements the functions of main engines for air-conditioning, engine start, and emergency power supply. Thus, APU installation must be considered, even though it weighs about 0.5 % of the aircraft empty weight.

The next step is to determine the location of the power plant installation. In the case of a military jet-engined fighter, such important items as intake and exhaust nozzle are carefully studied. For the transport's purpose, two mounting locations are compared between the under wing mounting and the rear fuselage mounting. The two mountings have their own merits or drawbacks, which are summarized in the Figure 4.3/3. In addition, the ground clearance for the wing mounted engine must be maintained to avoid a collision with the runway.

The number of engines can be selected for a integrated configuration by "rule of thumb" and the designer, in the parametric study, can get the engine of suitable thrust which can be obtained with the total required thrust divided by the number of engines. If the designer fails to find a suitable engine, he must backtrack to the configuration phase, re-select the number of engines, and a new thrust is re-estimated, thus finding correctly the suitable engines. This procedure will be explained in more detail in Chapter 5.

For preliminary design purposes, it is sufficient to estimate the thrust at take-off and cruise, together with the specific fuel consumption. These processes are implemented in the parametric study.

At this stage the task is to select from among the existing engines, the proper one, i.e., which produces the required thrust for low specific fuel consumption.

Since the jet engine was developed, there have appeared many derivatives such as the Turbojet, Turbofan, Turboprop, and so on. One such derivative is the Propfan (or Unducted Fan), which has been developed to overcome the noise and fuel consumption problems caused by "oil price hike" and it is newly being equipped into transport such as MD - 80.

Here, it is assumed that three kind of engines such as Turboprop, Turbofan, and Prop-fan are well suited for the transport category. Actually many types of engines are stored in this expert system, DESAID. In this process, if designer fails to find a suitable engine from among the existing engines, he must revise the engine number in the configuration phase and re-estimate the thrust, thus correctly choosing the suitable engine.

As described in Reference [7], the turbofan engine assumes good characteristics such as a good propulsive efficiency and low fuel consumption in the high subsonic range, in comparison with the other types of engine.

4.4.6. Tailplane Design

The conventional tailplane is composed of a vertical stabilizer and horizontal stabilizer and plays an important role in aircraft control and stability.

The tailplane is used to give pitching, yawing, and consequent angular changes, thus controlling the aircraft and providing the satisfactory stability characteristics. The tailplane sizing during the preliminary stage is complicated since it influences and is influenced by many other components. Thus discreet consideration must be given to the aerodynamic characteristics coupled with the main wing, flying characteristics and dynamic behaviour, pilot's verdict, and distribution of masses and its variations to which various loading conditions are to be known. [2] [3] [6]

The magnitude of pitching and yawing moments produced about the centre of gravity of an aircraft is greatly governed by the size of tailplane surface, its efficient production of lift in the desired direction, and the distance of the tail surface from the centre of gravity of the aircraft. In fact, the subsequent stability and control analysis depends upon the tailplane sizing.

The first thing to arrange for in the tailplane design is to determine its shape, number and position, with respect to the horizontal stabilizer and vertical stabilizer as follows;

1. Three kinds of tailplane are usually met with i) the single(*) vertical stabilizer with a horizontal stabilizer mounted on the rear fuselage or vertical fin, ii) twin tail boom or twin tail, iii) the V-shape or butterfly shape tail.

2. To determine their associated locations, the designer must consider stability and control in the stall and post-stall condition, slip stream effects, jet efflux effects, and recovery from spins.

In the configuration phase, the feasible type, number, and locations of tail surfaces are determined. The next step for the preliminary purpose is to calculate the dimensions of tail surfaces planform according to the following considerations. [3] [4]

1. The Critical Mach No. of a tail shall be greater than that of wing and the swept angle of tail shall also be greater than that of wing.

2. With respect to the A.R.(Aspect Ratio) for civil transport, it is initially assumed to be 4.0 for the horizontal stabilizer and 2.0 for the vertical stabilizer respectively.

3. For the Taper Ratio, an initial estimate is 0.33.

4. The Incidence Angle is assumed to be 0.0, although the downwash from the wing might require some tail incidence.

5. An Airfoil section of the horizontal stabilizer shall be selected among the airfoil series and that of the vertical stabilizer must be symmetric.

6. If the engine is mounted at the rear fuselage, then the position of horizontal stabilizer will be probably on the fin (High T-tail). Thus the weight of horizontal stabilizer is increased up to 15 %, compared with that of the rear fuselage mounted horizontal stabilizer.

7. It is customary to use the following relationships for sizing the tailplane.

- $S_{H,T} = C_{H,T} * S_W * M.A.C._{wing} / l_{H,T}$

  $S_{V,T} = C_{V,T} * S_W * b_{wing} / l_{V,T}$

$S_{H,T}$ = Horizontal stabilizer area

(*) Either the rear fuselage mounted horizontal stabilizer or the vertical fin mounted horizontal stabilizer will be considered for this expert system.

$S_{V,T}$ = Vertical stabilizer area.
S_W = Wing Area.
b = Wing span.
M.A.C. Mean Aerodynamic Chord of Wing.

$$\text{M.A.C.} = 2/3 * C_r(\text{ Root Chord }) * (1 + T.R + T.R^2)/(1+ T.R)$$

T. R = Taper Ratio

$l_{H,T}$ = Distance from $\text{M.A.C}_{1/4}$ of horizontal stabilizer to $\text{M.A.C.}_{1/4}$ of wing

$l_{V,T}$ = Distance from $\text{M.A.C}_{1/4}$ of vertical stabilizer to $\text{M.A.C.}_{1/4}$ of wing

$C_{H,T}$(*) = Volume Coefficient of Horizontal Tail, initially assumed to be 1.10

$C_{V,T}$ = Volume Coefficient of vertical Tail, initially assumed to be 0.08

$l_{H,T}$ / M.A.C. is assumed to be 1.92 to 4.56 for the under wing mounted engine 3.91 to 4.81 for the rear fuselage mounted engine (Data collected from the reference [6]).

$l_{V,T}$ / M.A.C. is assumed to be 1.83 to 4.19 for the under the wing mounted engine 3.21 to 4.91 for the rear fuselage mounted engine.

8. The weight of a tail surface is some 2 % of the gross mass and is affected by the centre of gravity. The following equation applies to the transport airplane and to the business jets with the design dive speeds above 250 knots.

$$W_H=K_H*S_H(3.81*\{S_H^{0.2}*V_D\}/\{1,000*(\cos\Lambda_{1/2h})^{1/2})-0.287)$$

$$W_V=K_V*S_V(3.81*\{S_V^{0.2}*V_D\}/\{1,000*(\cos\Lambda_{1/2h})^{1/2})-0.287)$$

W_H : Horizontal tail Weight

W_V : Vertical Tail Weight

K_H : 1 for fixed incidence angle
1.1 for variable incidence angle

(*) The tail volume coefficient is a factor that can be determined from the comparison with the other airplanes of the similar type. [4]

K_V : 1 for fuselage mounted horizontal tail

$1+0.15*(S_H*Z_H/(S_V*b_V))$ fin mounted horizontal tail

V_D : Design Diving Speed in KEAS.

$/\backslash_{1/2}$: semi - chord swept angle.

If the composite material is to be used, 20 % weight saving is assumed.

4.4.7 Landing Gear Design

The undercarriage must be required to absorb both the horizontal impact and the vertical impact upon touching the ground during taxiing, lift-off, and touch down so that no components of the aircraft collide with the ground. Further it shall be designed to be adapted to the airfield's load capacity. After the type, number, and position are selected, the next step is to estimate the overall size and weights.

Even though it seems that the dimensions of the undercarriage are less than those of the wing or fuselage, it is not an accessory but an integral part of the structure. In the preliminary design stage, designers do not have to investigate the details of undercarriage and to design the related hydraulic equipments. However, the general arrangements such as wheel positions with respect to airframes and kinematics of retractions must be carefully tailored and the following functional requirements will be considered to generate an undercarriage layout in the preliminary design phase. [3]

1. During the Take-off rotation, lift-off, landing flare-out and touch down, the wheels only should be in contact with the ground with adequate clearance between the runway and the aircraft components such as wing and its tips, propeller and rear fuselage.

2. The tire inflation pressure and landing gear configuration shall comply with the bearing capacity of airfields where the aircraft is to be operated.

3. The undercarriage should absorb the vertical landing impact loads when the touch-down rate (sink speed) is 10 fps (feet per second) for an transport.

4. The braking force should be sufficient, yet avoid instabilities such as canting or ground looping happening due to a landing in a crosswind and taxiing at high speed.

5. The structural elements at the attachment points for the undercarriage should be strong enough to bear the load exerted by the undercarriage and an internal space shall be arranged for a suitable retraction. To stow the undercarriage under the wing, the commonly used type is the Yehudi type, which has the extended root chord and the enlarged surface by the connection between the trailing edge of root chord and that of chord separated a little bit from the root chord. This type can be seen on the wing fuselage attachment of the modern transports such as B-747, 767, A300, etc.

6. The weights of landing gear fall within the range of 0.03 to 0.05 times the maximum Take-off weight and 0.3 to 0.5 times the structural wing weight. The typical methods for estimating the Landing - Gear Weight of transport were developed by General Dynamics, E. Torenbeek, Cessna, etc. [3] [6]

   $$W_{u/c} = 62.61 \; (W_{TO} / 1000)^{0.84}$$

7. In the undercarriage layout, it must be checked that all the geometric clearances and tip-over, spray angle, pitch and bank during Take-off and Landing, and turn radii are within the proper limit. (*)

In conclusion, the summarized advantages and disadvantages of various undercarriage layout were described in the figure 4.4.3.1/1.

4.4.8 Weight Analysis

The primary goal of aircraft design is to minimize the aircraft weight since an increase in weight in one part results in weight growth in other parts and thus leads to snowballed weight growth. Therefore the weight prediction during the initial design phase must be accurate and all the effects, resulting from the general aircraft layout and geometry, must be accounted for.

The weight groups of an aircraft can be divided as shown in the figure 4.4.8.1/1. The weight items to mainly consider in this thesis are the airframe structural weight, empty weight, fuel weight, payload, maximum take-off weight, and the maximum landing weight, which are already known from the parametric study. Here they will be cross-checked as follows;

(*) This calculation will not be covered here but many references cover the methods of this calculation. [3] [6]

1. The sum of structural weights calculated in component designs (e.g., wing, fuselage, tail surface, engine, and undercarriage) must be close to an empty weight of the parametric study within a certain range of error, more or less 10 %.

2. Whether the Maximum take-off weight from the parametric study is close to the sum of structural weights, fuel weight, payload, and so on.

3. Of course, the weight of miscellaneous items such as avionics, electrical system, hydraulic system, etc., should be considered but omitted for the similar reasons described in the previous section.

In the weight prediction, the designer can rely on such various approaches as the empirical comparison as shown in the Figure 4.4.8.1/2, the following formulae, and theoretical formulae. However, the empirical comparisons are too broad and not exact.

On the other hand, the theoretical methods are too complex for the initial design. Thus an empirical formula is commonly used and it is based on the statistical weight data, which can be expressed in the following exponential form. [3]

$$\text{Weight} = C1 * (\text{Variable } 1)^{A} * (\text{Variable } 2)^{B} \text{ ---- } + C2$$

: Variable 1 & 2 are such design variables as wing span, fuselage weight, sweep angle, number of passenger, etc.

: C1 & C2 are constant for the equation.

They are particularly suitable for the initial design study to allow a range of parametric studies. The typical equations are well described in the reference [6, volume 5]. The author, however, followed the Cranfield method when applicable.

It is worth mentioning here that Aircraft companies and manufacturers have invested considerable effort to develop their own aircraft weight estimate methods for the preliminary design purpose and hence are very reluctant to publish outside. However, there are some methods available for use in the initial design stage and these methods are described in References [3],[4], and [6].

As this research just needs to decide whether the weight estimates fall within the appropriate range, the weight of each component is estimated first and compared with the statistical values gathered from the current Jet Transport.

4.4.9 Cost Analysis

4.4.9.1 Introduction

In addition to the completion of component designs, it is essential to estimate the aircraft operating cost so that the configuration might be decided or revised depending upon the result of cost estimates.

With regard to <u>cost</u>, it can be defined as the total amount of resources (i.e. US Dollar or UK pound) needed to manufacture an airplane. The <u>price</u> of an airplane is the amount paid for the aircraft and the profit is the deduction from the price by the cost. A total airplane program is an evolutionary process from initial design to manufacturing, operation, and disposal, the whole process of which is called as an *<u>Aircraft Life Cycle</u>*. *An <u>Aircraft Life Cycle Cost</u>* is the total cost incurred during the aircraft life cycle. Generally, the aircraft life cycle cost can be broken down into four cost sources. [6]

1. *The Research, Development, Test, and Evaluation Cost*(C_{RDTE})
2. *The Acquisition Cost is a manufacturing cost* (C_{MANU}) *plus manufacturer's profit* (C_{PROF}).
3. *The Operating Cost* (C_{OPS}).
4. *The Disposal Cost* (C_{DISP})

The life cycle cost is the summation of all the above costs and the biggest is the Operating Cost while the second biggest is the Acquisition Cost, the third C_{RDTE}, and the smallest is the Disposal Cost. [5] [6]

The Figures 4.4.9.1/1 and 4.4.9.1/2 show that the conceptual and preliminary designs have significant leverages affecting the aircraft life cycle cost in that the aircraft life cycle cost is locked in during these two phases. Thus, it is required for the designer to conduct a preliminary cost analysis in the initial design stage in order to find out the minimum life cycle cost.

One thing important to bear in mind is the cost escalation factor due to inflation, in consideration of the fact that an aircraft development program may take many years. Thus, it is customary to scale cost data from 'then-year' to another with the cost escalation factor (CEF) as follows;

$$\text{Cost}_{19xx}\ \text{year} = \text{Cost}_{19yy}\ \text{year}\ (\text{CEF}_{19xx} / \text{CEF}_{19yy})$$

Among the above costs, due to the voluminous requirements to implement a cost program, the author regarded it to be reasonable to consider only the Direct Operating Cost.

4.4.9.2 AEA Cost Model

The appropriate methods for estimating the direct operating cost of transport category aircraft have been investigated in the Boeing Cost Model, the Cranfield Research Study, and AEA (Association of European Airlines [27] Model). The Author found it adequate to adopt the AEA cost model for this research.

Thus the Direct Operating Cost estimate is influenced by the following factors such as

1. Ranges : Flight Range in specification
2. Annual Utilization : This will be calculated according to the average per flight block time. The definition of block time is the time lapses between the start of taxi - out and the completion of taxi-in.
3. The Gross weight(W_G), Fuel Weight(W_f), and payload(W_p) which are estimated in the parametric study.
4. Number of Engines : 2 or 3 or 4
5. By-pass Ratio
6. Static Take-off Thrust at sea level : T_O

This method calculated the direct operating cost in terms of seat-mile costs and aircraft-mile costs and they were described in Appendix I.

Component	Component Type	Component Number	Component Position
Fuselage	1. Circular 2. Double-Bubble	. One	. Center-Line
Wing	1. Backward Sweep 2. Forward Sweep	. One	. Low wing . High Wing
Engine	1. Turbofan 2. Propfan 3. Turboprop	. One . Two . Three . Four . Five . Six	. Under-Wing Mounted . Rear Fuselage Mounted
Horizontal Tail	1. Back(ward) Sweep	. One	. Rear Fuselage Mounted
Vertical Tail	1. Back(ward) Sweep	. One	. Rear Fuselage Mounted . Fin Mounted
Landing-Gear	1. Retractable	. Three . Four	. NoseBody-Wing . NoseBody-Wing-Fuselage

Figure 4.3/1 Classification of Component

Parameters to be effected	**High Wing**	**Mid Wing**	**Low Wing**
Interference Drag	Poor	Good	Poor
Dihedral	Negative	Neutral	Positive
Visibility	Good	Good	Poor
Loading Unloading	Easy	Easy	Need Stairs

Figure 4.3/2 The Summary of Effects due to Wing Location [6]

CRITERION	WING-MOUNTED	FUSELAGE-MOUNTED
Ground Clearance	Problem Possible	GOOD
Internal Noise	Fair	Good
Acoustic Fatigue	Flap/Wing	Fuselage
Crash Safety	Good	Possible Problem
Propulsive Efficiency	Good	Good If well positioned
Longitudinal Stability	Good & Delay Tip Stall	Loading Problem Short Tail Arm and Tip Stall
Asymmetric Thrust	Poor	Good
Weight	Good Wing bending & Torsion Relief	Heavy Tail, Heavy Fuselage
Maintenance	Good	Need Assistance
Wing Aerodynamic Efficiency	Possible Flap & L. E. Cutouts	Very Good
Fuel Feed	Good	Long Pass Lines
Anti-Ice	Easy to use	Long Line Duct through Cabin

Figure 4.3/3 The Compared Features between Wing Mounted Engine and Rear Fuselage Engine [2]

DESIGN ACTIVITY	*DESIGN CHARACTERISTICS*
1. Parametric Study	1. Landing Performance 2. Take-off Performance 3. Cruise Performance 4. Size Matching (Wing Loading, Thrust Loading) 5. Resulting Parameters For Sizing Aircraft
2. Wing Design	1. Airfoil Selection 2. Drag-Rise 3 Dimensional 3. Flap Effectiveness 4. Aeroelastics 5. Tip Stall Check 6. Wing-Weight => Wing Planform
3. Fuselage Design	1. Length / Diameter ratio 2. Fuselage Length 3. Fuselage Cross Section
4. Engine Design (selection)	1. Engine Thrust Estimate 2. Engine Selection
5. Tail (Horizontal, Vertical)	1. Tail Position 2. Tail Planform
6. Weight Analysis	1. Weight per Component 2. Weight Overall
7. Cost Analysis	. Direct Operating Cost

Figure 4.4.1/1 Design Activity for Design Phase

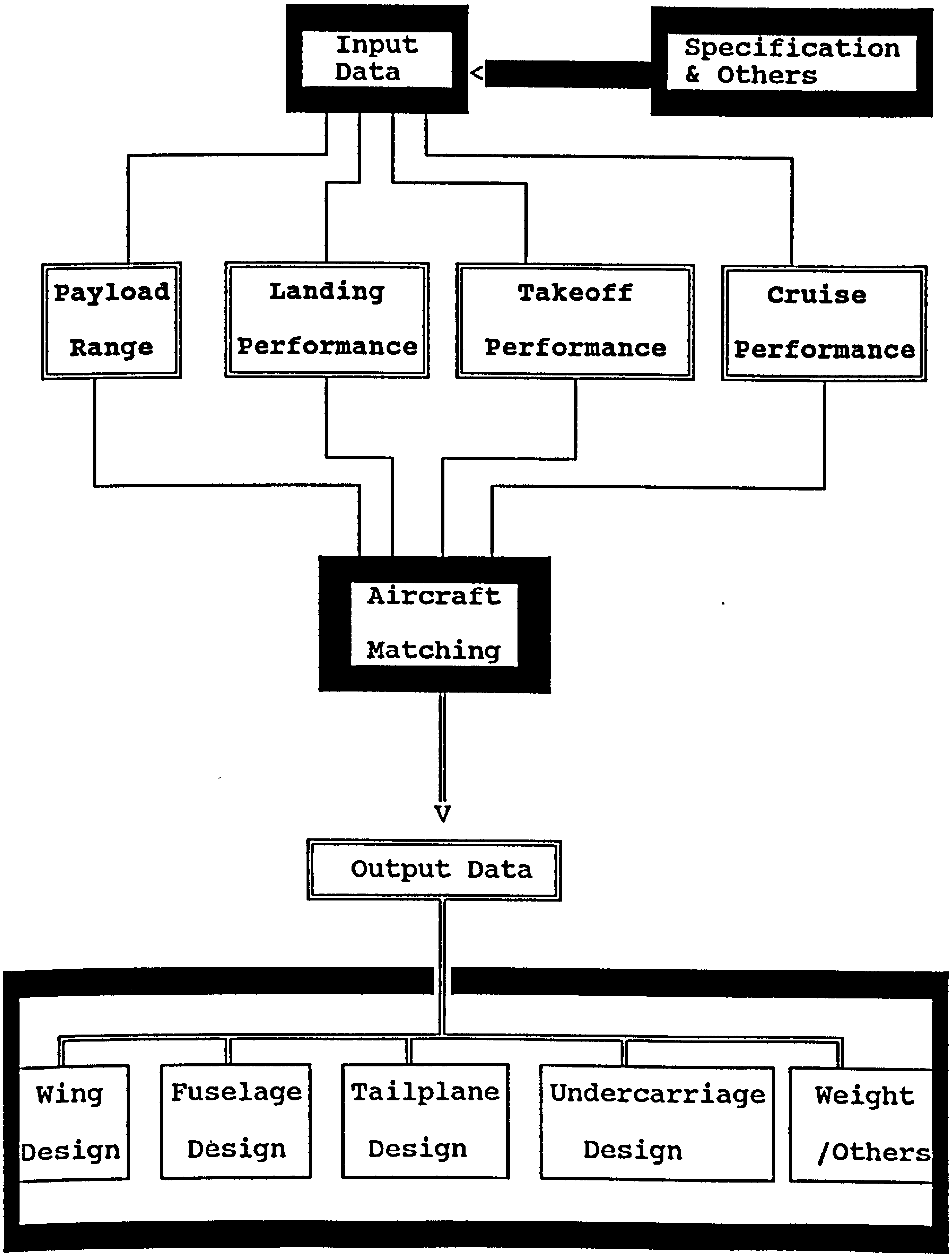


Figure 4.4.2.2/1 The Parametric Study Procedure

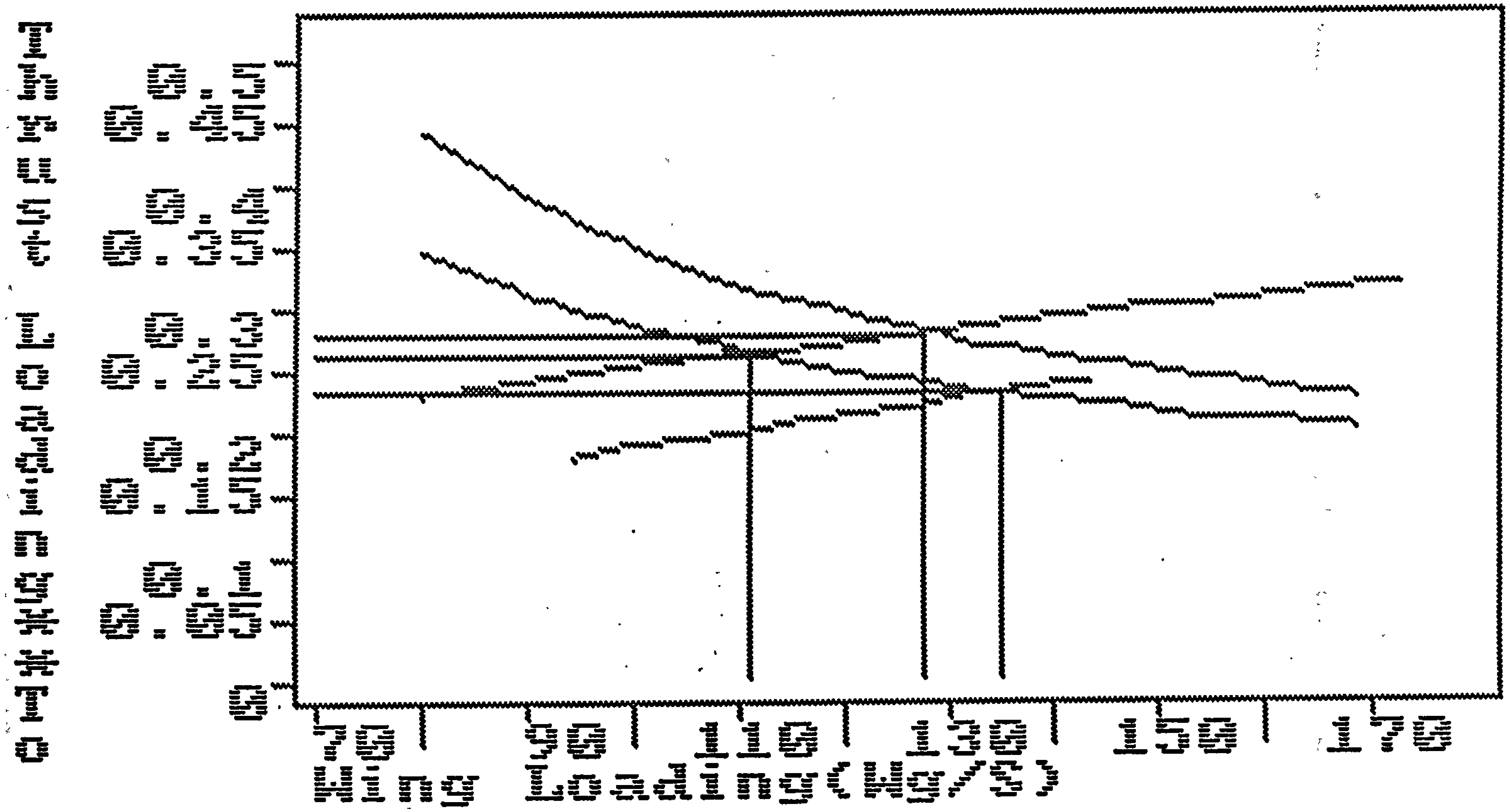


Figure 4.4.2..2/2 The Example of Finding Matching Points

Parameters to be effected	Swept Forward	Swept Backward
Lift Curve	Low	Low
Pitch in Low speed	High	High
Ride through Turbulence	Good	Good
Asymmetric Stall	Best	Poor
Lateral Control at stall	Best	Poor
Onset of Compressibility	Low	Low
Weight of Wing	High	High
	High Aspect Ratio	Low Aspect Ratio
Induced Drag (CL^2/Phi*A*e)	Low	High
Lift Curve	High	Low
Pitch Attitude	Low	High
Ride in Turbulence	Poor	Good
Wing Weight	High	Low
Wing Span (b^2/S)	Large	Small
	Low Thickness	High Thickness
Wing Weight	High	Low
Wing Drag		
Subsonic	Low	High
Supersonic	Normal	High

Fuel Volume	small	Large
Maximum Lift	Poor	Good
	Positive Dihedral	Negative Dihedral
Spiral Stability	increase	decrease
Dutch Roll	decrease	increase
Ground Clearance	Good	bad
	Large Incidence	Small Incidence
Cruise Drag	High	Low
Cockpit Visibility	Good	Watch Out
	Tricycle	Bicycle
Groundloop	Stable	Stable w.r.t C.G.
Visibility	Good	Good
Weight	Medium	High
Steering	Good	Normal
Take-Off Rotation	Good	Need care

Figure 4.4.3.1/1 The Summary of Effects due to Configuration Parameters [6]

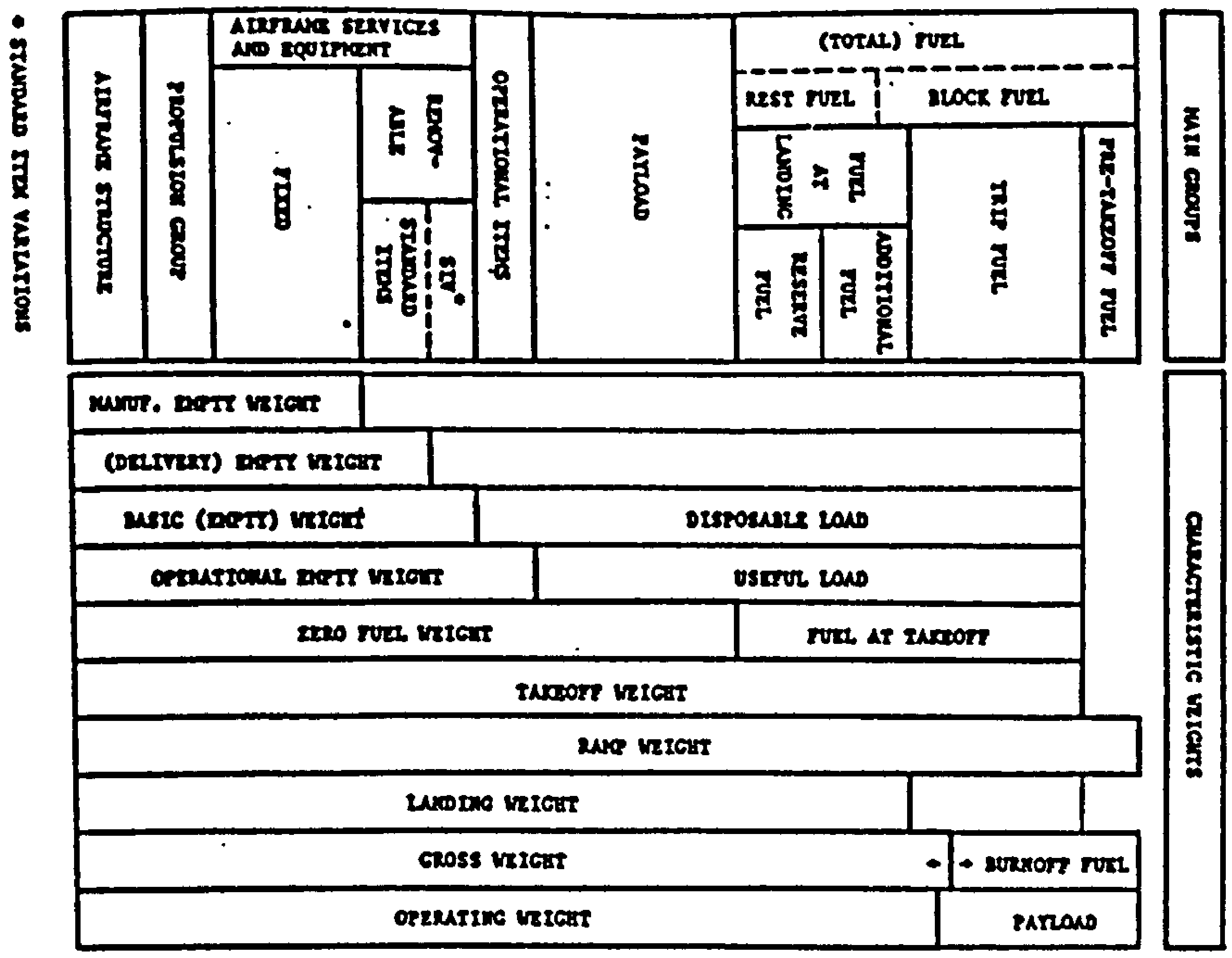


Figure 4.4.8.1/1 The Weight Group and Classification (p269, [3])

AIRPLANE CATEGORY AND TYPE		MTOW	WING GROUP		TAIL GROUP		FUSELAGE GROUP		LANDING GEAR		SURFACE CONTROLS		NACELLE GROUP	
		10^3 lb	10^3 lb	%	10^3 lb	%	10^3 lb	%	10^3 lb	%	10^3 lb	%	10^3 lb	%
2 ENGINES	VFW-Fokker 614	40.981	5.767	14.1	1.121	2.74	5.233	12.8	1.620	3.45	0.745	1.82	0.971	2.37
	Fokker-VFW F-28/1000	65.000	7.330	11.3	1.632	2.46	7.043	10.8	2.759	4.24	1.387	2.13	0.834	1.28
	F-28/2000	65.000	7.347	11.3	1.632	2.46	7.649	11.8	2.759	4.24	1.400	2.15	0.834	1.28
	F-28/5000	70.800	8.223	11.6	1.632	2.31	7.043	9.95	2.759	3.90	1.665	2.35	0.849	1.20
	F-28/6000	70.800	8.244	11.6	1.632	2.31	7.649	10.8	2.789	3.94	1.674	2.36	0.849	1.20
	BAC 1-11/300	87.000	9.643	11.1	2.369	2.72	9.713	11.2	2.865	3.29	1.481	1.76	••	-
	1-11/400	87.000	9.670	11.1	2.419	2.78	9.743	11.3	2.899	3.33	1.207	1.39	••	-
	McD. Douglas DC-9/10	91.500	9.470	10.3	2.630	2.87	11.206	12.2	3.660	4.00	1.264	1.38	1.417	1.55
	Boeing 737-100	97.800	9.968	10.2	2.700	2.76	12.380	12.7	3.687	3.77	1.589	1.62	•••	-
	737-200	100.000	10.613	10.6	2.718	2.72	12.108	12.1	4.354	4.35	2.348	2.35	1.392	1.39
	Aerospat. Caravelle XIR	110.230	14.735	13.4	1.957	1.77	11.570	10.5	5.110	4.63	2.063	1.87	1.581	1.43
	Airbus A300B/2	304.000	44.131	14.5	5.941	1.95	35.820	11.8	13.611	4.47	5.808	1.94	7.039	2.32
3 ENGINES	H. Siddeley 121-1C	115.000	12.600	11.0	3.225	2.80	12.469	10.8	4.413	3.84	1.792	1.56	••	-
	121-1E	134.000	13.462	10.0	3.341	2.49	13.328	9.95	5.073	3.79	1.689	1.26	••	-
	Boeing 727-100	161.000	17.764	11.0	4.133	2.57	17.681	10.9	7.211	4.48	2.996	1.86	3.864*	2.40
	727-100C	160.000	17.492	10.9	4.142	2.59	20.044	12.5	6.860	4.29	2.957	1.85	3.839	2.40
4 ENGINES	Boeing KC-135	297.000	25.251	8.50	5.074	1.71	18.867	6.35	10.180	3.43	2.044	0.69	2.575	0.87
	707-121	246.000	24.026	9.76	5.151	2.09	20.061	8.15	9.763	3.97	2.044*	0.83	4.639	1.89
	707-320	311.000	29.762	9.57	5.511	1.77	21.650	6.96	12.700	4.08	2.400*	0.77	4.497	1.45
	707-320C	330.000	32.255	9.77	6.165	1.87	26.937	8.16	12.737	3.86	3.052	0.92	4.183	1.27
	707-321	301.000	28.647	9.52	6.004	1.99	22.129	7.35	11.122	3.70	2.408	0.80	5.119	1.70
	720-022	203.000	22.850	11.3	5.230	2.58	19.035	9.38	8.110	4.00	2.430	1.21	4.510	2.22
	747-100	710.000	86.402	12.2	11.850	1.67	71.845	10.1	31.427	4.43	6.982	0.98	10.031	1.41
	747-200B	775.000	92.542	11.9	11.842	1.53	72.053	9.30	32.693	4.22	7.073	0.91	10.136	1.31
	McD. Douglas DC-8-10	273.000	26.235	9.61	4.740	1.74	21.495	7.87	10.185	3.73	2.000*	0.73	3.505	1.28
	DC-8-55	328.000	34.759	10.6	4.889	1.49	22.248	6.78	11.255	3.43	2.253	0.69	4.685	1.43
	BAC VC-10-1101	312.000	34.672	11.1	6.958	2.23	25.113	8.05	10.489	3.36	•••	-	••	-
	G. Dynamics 880	184.500	17.669	9.58	4.247	2.30	13.699	7.42	6.203	3.36	•••	-	3.685	2.00
	990	253.000	26.871	10.6	5.326	2.11	16.673	6.59	8.718	3.44	•••	-	6.772	2.68

• estimated   •• included in other items   ••• no data available

Figure 4.4.8.1/2 The Analysis of Weight Group ([7])

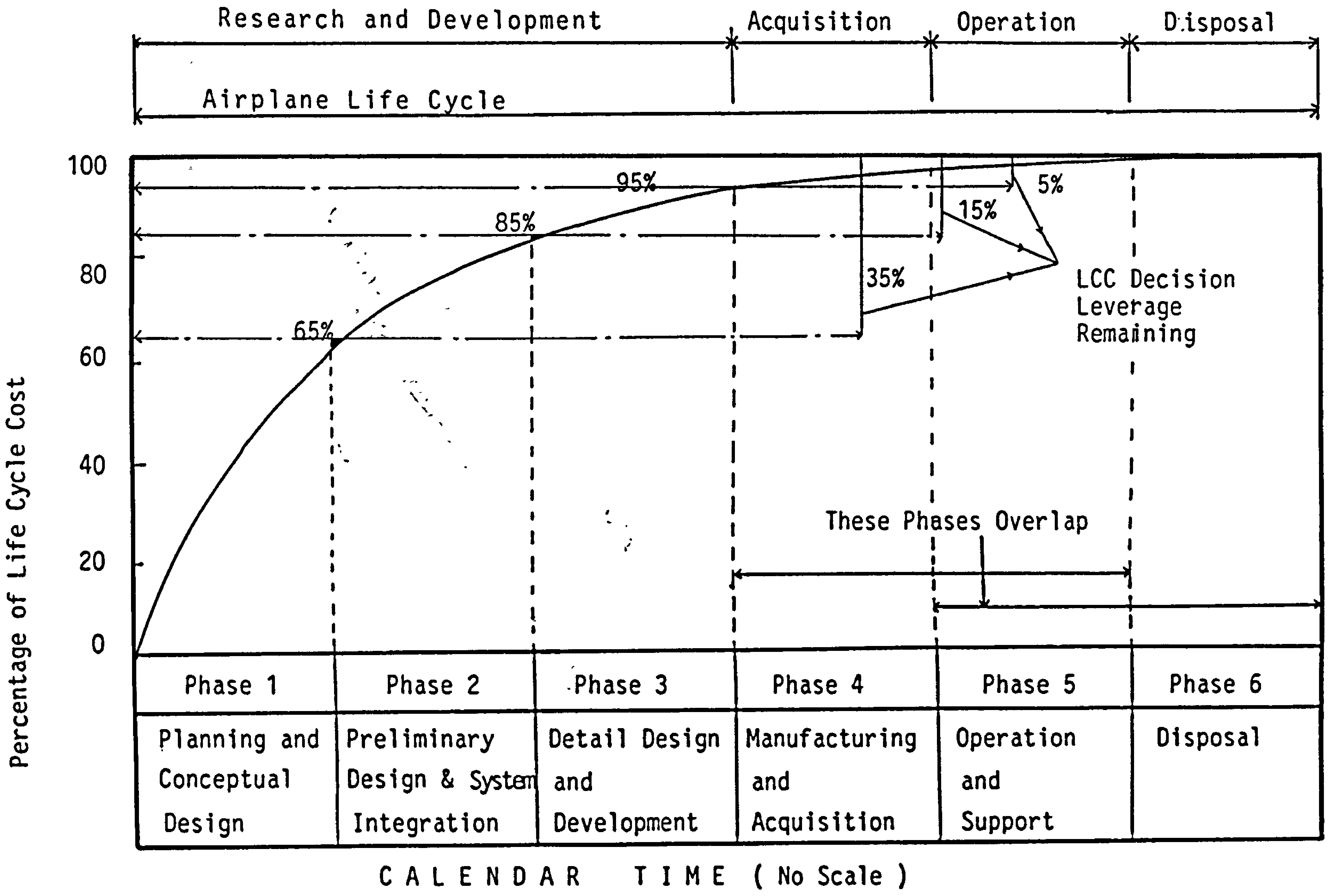


Figure 4.4.9.1/1 The Impact of Airplane Program Phases on Life Cycle Cost (p10, [6])

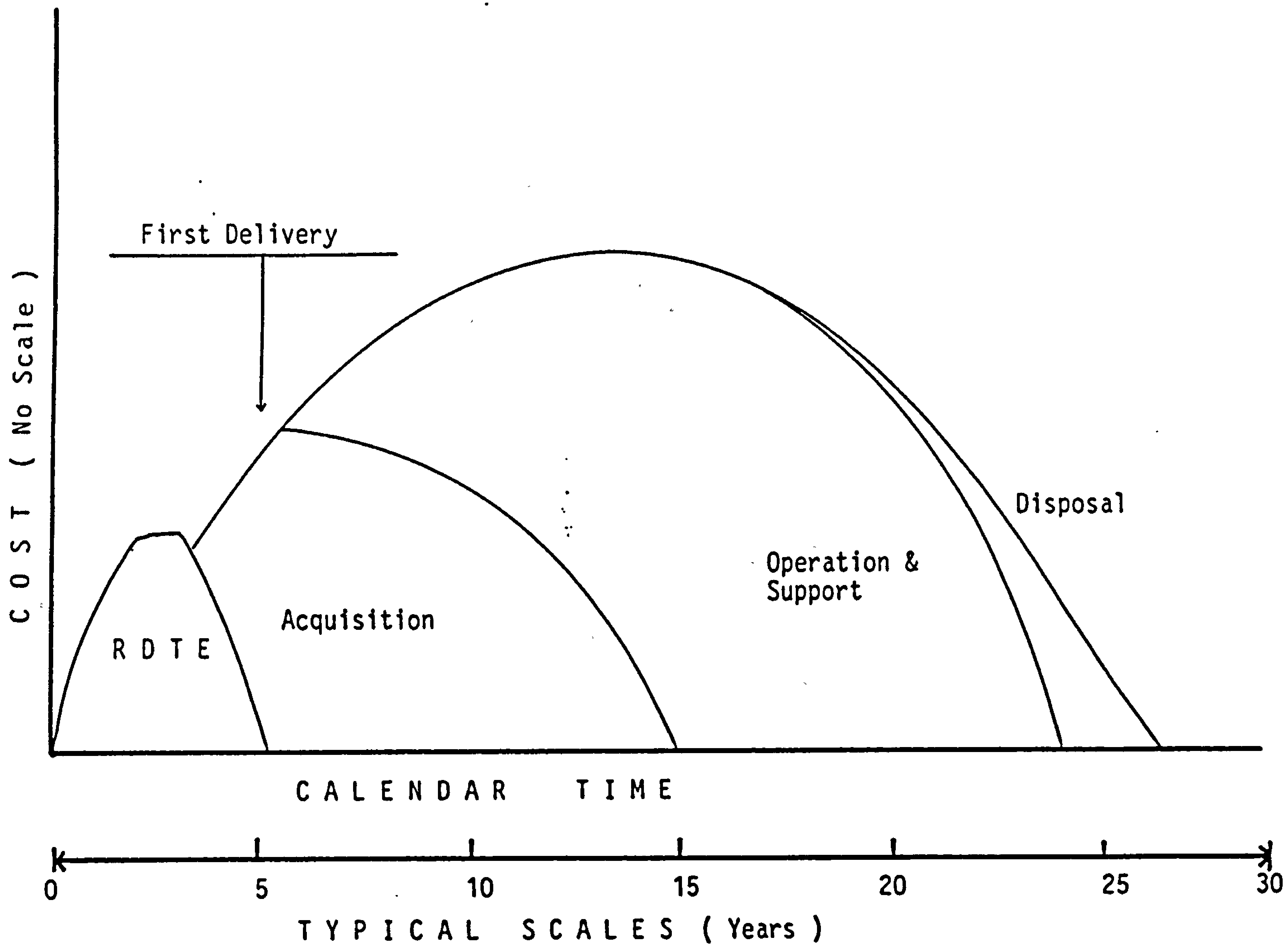


Figure 4.4.9.1/2 Schematic Representation of Life Cycle Cost History (p13, Vol. 8, [6])

CHAPTER 5

PROBLEM STRUCTURING AND STRATEGY FOR SOLUTION

" *Design as prototype refinement involves working within the constraints of a particular class of designs - ... The second design category, prototype adaptation, involves extending the boundaries of a particular class of designs -... In aircraft design, for example, we elect to place the jet engine above the wings so that the exhaust gases pass over the wing surface and increase lift, --.. The third category, prototype creation, or conceptual / original design, is where totally new prototype emerge, as in the design of the first airplane....* " -- page 33, [49]

CHAPTER 5 PROBLEM STRUCTURING AND STRATEGY FOR SOLUTION

5.1 Complexity of Configuration Design Problem

As described in previous chapters, the problem of aircraft configuration design is complex in that a sophisticated result is drawn from basic concepts based on set of requirements. It invloves an integration of the required components, and a design activity to estimate the size of synthesized configuration and its components, as shown in figure 5.1/1

To set up the basic concept based on the requirements, the following considerations must be examined. This stage is here called *"Set-Up"* phase.

1. What is the purpose of the aircraft ?

   . *Civil or Military*

2. How will the aircraft be categorized ?

   . *Transport or Fighter, etc.*

3. Within which Speed Range will the aircraft operate ?

   . *Subsonic, Transonic, Supersonic, or Hypersonic*

4. Which Design Avenue will the aircraft adopt ?

   . *Conventional or Unconventional*

5. What method of Take-Off and Landing concept will the aircraft employ ?

   . *Conventional Take-Off and Landing (CTOL)*
   . *Short Take-Off and Landing (STOL)*
   . *Vertical Take-Off and Landing (VTOL)*
   . *Short Take-Off and Vertical Landing (STOVL)*

With the appropriate items set-up above and in order to integrate the required components into a whole configuration, the designer must classify and decide the type, number, and position of certain aircraft components such as Fuselage, Wing, Engine, Tail and Undercarriage, as illustrated in figures 2.2.1/1 and 4.4.1/1. This stage is here called *"Configuration Phase"*.

Next task is to implement the actual design processes which involve operating with the aircraft design disciplines, as shown in Figure 4.3/2 and Appendix I. This stage is here called *"Design Phase"*.

5.2 Problem Structuring

To structure the overall configuration design problems within the above three phases, the author developed the following three structures, *namely, Tree Structure, clustering, and Layer-Node Concept,* and investigated them to find one suitable for this system development(*).

1. Tree Structure

Tree structure has been commonly used by many A.I. researchers during the early stage of A.I. development, to find a path to a solution (i.e., legal chess move).

Considering the application of this concept to this design problems as shown in Figure 5.2/1, the paths needed to connect the nodes of design processes were calculated as follows;

Purpose (2) *
Category (5 per Category) *
Speed Range (4) *
Concept (2) *
T-O and Land (4) *
Configuration Component (6) *
Type (At least 2 per Component) *
Number (At least 2 per Component) *
Position (At least 2 per Component) *
Parametric Study (5 process per Aircraft) *
Design Activity (At least 3 per Component Design)

(*) This system is called *D E S A I D* (The Development of an Expert System for Aircraft Initial Design)

The total number of paths needed to cover the design problem is close to about 240,000. To represent them for programming purposes would require, at least, 5 Mega-bytes (240,000 paths * 20 byte per path). This could not be processed with real memory storage of present A.I. computer which is based on DOS system. To make the matter worse, that number of paths gives rise to a *Nodal Explosion* which is time-consumption and makes the problem solving more complex. This structure is unrealistic and impossible to organize and was discarded.

2. Clustering

Clustering represents an alternative for reducing the paths in the tree structure and this concept, as shown in the figure 5.2/2, is to group together the nodes belonging to the same process and to cluster them to the process. Even if the required paths can be reduced compared with Tree Structure, it still required a great number of paths not only among the processes but also between a main process and its clustered nodes. Thus, this was also discarded.

3. Layer - Node Concept

If the processes and the nodes in Figure 5.2/1 are given the priorities while they are being processed, they can be represented hierarchically. Thus, the nodes relating to the same process could be laid out as shown in figure 5.2/3 and in this situation pathways are no longer required. Instead, we now need relationships between the process and its nodes. Also, the priorities could be given, if required either among processes or among nodes. The search is not so complex in that the nodes in the design process can be searched naturally if they are also given constraints. For example, if we are to design a civil transport whose type in specification is similar to existing ones and whose cruising Mach number is less than 0.85, then the nodes to be selected are

1. *Civil Nodes in Purpose*
2. *Transport Nodes in Category*

3. *Subsonic Node in Speed because M_{cr} is less than 1.0*
4. *Conventional Node in Concept because the type is similar to existing one.*
5. *CTOL in T-O and Land because the type is similar to existing one.*
6. *The Fuselage's type, number, and position are circular, one, and center-line, respectively because the type is similar to existing one.*

Therefore, this concept was adopted. In principle, the design process is structured into "Layers" with nodes which are given constraints. Its characteristics can be described as follows;

1. An order of precedence is set among Layers and also among Nodes if all the nodes relating to a given layer must be processed.

2. Some layers have one node, while all the nodes of other layers must be processed completely.

3. Following Layers are decomposed into sublayers, as shown in Figure 5.2/3.

   . Component Configuration --> Type, Number, Position.

   . Design Activity --> Design Template and its subprocesses.

5.3 Knowledge Base and Inference Engine

As shown in Figure 5.3/1, to develop a expert system for solving configuration design problems structured in the Layer_Node concept, it is required for a designer to :

1. Construct the Knowledge Base by

   .. Representing all the design knowledge obtained from the analysis in Chapter 4 and Appendix I.

   .. Converting the knowledge into A.I. Programming Language Expression for execution.

2. Devise an Inference Engine which deals with the knowledge and searches for a solution by using A.I. techniques.

5.3.1 Knowledge Base

Among the knowledge representation techniques such as rules, semantic nets, frame, and object oriented methods, the rule expressions were adopted.

The semantic nets was originally developed for the psychological models of human memory. Instead, the frame is a concept similar to the semantic nets. Considering the above fact and the decision to use PROLOG as explained in Chapter 2, the Object - Oriented method was discarded because SMALLTALK was not adopted and hence the Rule Expression was adopted.

Thus, the following rules of design knowledge and its equivalent PROLOG expression can be expressed. (*)

Rule

- If maximum cruising speed in the specification is less than 0.9,
  then LAYER is SPEED and NODE is SUBSONIC.
- If landing distance is greater than 3,000 ft and less than 6,000 ft, and take-off distance is greater than 5,000 ft and less than 12,000 ft,
  then LAYER is CONCEPT and NODE is CTOL
  (Conventional Take-off and Landing)

PROLOG

-
  ```
  layer_node(speed,subsonic):-
                user_r(mmax,_Mcr),0<_Mcr,_Mcr<=0.9.
  ```

(*) These rules are parts of the total rules and the total rules and their PROLOG expressions were in the APPENDIX II and III.

```
layer_node(takeoff_land,ctol):-
          user_r(land_d,_Dist),user_r(t_o_d,_t_o_d),
          3000<_Dist,_Dist<=6000,
          5000<=_t_o_d,_t_o_d<12000.
```

5.3.2 Inference Engine : Layer and Node Concept

1. *Search*

As shown in Figure 5.3.2/1, each layer has a number of nodes, one or all of which must be selected for execution. All the layers are in the order of precedence during execution. The nodes of each layer, if all of them must be implemented, have also an order of precedence. The related expressions are as follows;

SELECTION TYPE

- SELECTION of LAYER VEHICLE is ONE.
- SELECTION of LAYER PURPOSE is ONE.
- SELECTION of LAYER CATEGORY is ONE.
- SELECTION of LAYER SPEED is ONE.
- SELECTION of LAYER CONCEPT is ONE.
- SELECTION of LAYER TAKEOFF_LAND is ONE.
- SELECTION of LAYER CONFIGURATION_COMPONENT is ALL.
- SELECTION of LAYER TYPE is ONE.
- SELECTION of LAYER NUMBER is ONE.
- SELECTION of LAYER POSITION is ONE.
- SELECTION of LAYER DESIGN_ACTIVITY is ALL.
- SELECTION of LAYER DESIGN_TEMPLATE is ALL.

LAYER PROCESS PRIORITY

- LAYER VEHICLE must be followed by LAYER PURPOSE.
- LAYER PURPOSE must be followed by LAYER CATEGORY.
- LAYER CATEGORY must be followed by LAYER SPEED.
- LAYER SPEED must be followed by LAYER CONCEPT.
- LAYER CONCEPT must be followed by LAYER TAKEOFF_LAND.
- LAYER TAKEOFF_LAND must be followed by LAYER CONFIGURATION_COMPONENT.

- LAYER CONFIGURATION_COMPONENT must be followed by LAYER DESIGN_ACTIVITY.
- SUBLAYER TYPE must be followed by SUBLAYER NUMBER.
- SUBLAYER NUMBER must be followed by SUBLAYER POSITION.

The summary of expressions and their related PROLOG expressions were shown in Appendix II and III respectively. Then, in the structure described in the Figure 5.2/3, each layer must select and execute one node , if layer_type is select_one (depth_first search). Otherwise, every node must be selected if layer_type is select_all (breadth_first search). By blending the depth first search and the breadth first search, the 'combined search' strategy was adopted.

2. *Inference Chain*

With respect to the Inference Chain, between the forward chaining and the backward chaining, the former was used in finding the next node or arriving at the conclusion as shown in figure 5.3.2/1, whilst the latter was used in showing the explanation why such a conclusion was arrived at, as shown below and in figure 5.3.2/2.

1. *Forward Chaining : Figure 5.3.2/1*

If the wing's position is low wing and the engine position is rear fuselage mounted,

|
V

Then the horizontal tail position is vertical tail mounted (or fin mounted).

In Prolog Expression,

```
sublayer_node(horizontal_tail,position,
              vertical_tail_mounted) :-

          sublayer_node(wing,position,low_wing),
          sublayer_node(engine,position,
          rear_fuselage_mounted).
```

2. *Backward Chaining : Figure 5.3.2/2*

From the above rule and prolog expression, the question "*how was the vertical tail mounted position selected ?*" can be shown in the explanation facility as "*because the low wing was selected and the engine position was rear fuselage mounted.*".

3. *Control Mechanism (Inference engine)*

As shown in Figure 5.3.2/3, the logic for controlling the **layer_node structure** starts from the **Set-up phase**, through the **Configuration phase**, and finishes at the **Design phase**.

3-1. The **Set-up phase** has the following six(6) layers and those layers are implemented according to the following order of precedence.

1. Vehicle
2. Purpose
3. Category
4. Speed
5. Concept
6. Take-off and Land

Each layer has its **'select' type(*)** which is **one**. This means that the system shows all the possible nodes of which the constraints satisfy the specification and design conditions. From them, the designer selects 'one' node and executes only the node, which means the execution of its layer.

If the system commences execution, the system shows first the layer **vehicle**'s nodes (i.e., **aircraft and spacecraft**) for the designer's choice and the selected node is marked by the system as '**selected (vehicle, aircraft)**'.

Likewise, the remaining layers are then executed in the same manner as the layer 'vehicle', as shown in Figure 5.3.2/3.

(*) The Turbo Prolog expressions of layer types are

1. **"layer_type(_layer,select_one)".**
2. **"layer_type(_layer,select_all)".**

3-2. In **Configuration Phase** next to the execution of 'Set-up' phase, the layer to be executed is **'component'** whose type is **'all'**. The **'select' type 'all'** means that according to the order of precedence the system shows (for designer's choice) the possible nodes of which the constraints satisfy the specification requirements and design conditions.

1. The designer must chose a node and implement the node. Then the designer must implement all the remaining nodes according to the order of precedence.

   The 'component' layer has the following six(6) **main nodes** and the number means the order of execution.

   1. Fuselage
   2. Wing
   3. Engine
   4. Vertical Tail
   5. Horizontal Tail
   6. Undercarriage

2. Each main node has its sublayers (i.e., **component's type, number,** and **position**) whose 'select' types are 'one'. This 'select' type 'one' has the same meaning as 'one' in the 'Set-up' phase. The sublayers are also executed in the order of 'type', 'number', and 'position'.

   The system shows, at first, **main node 'fuselage'** and **its 'sublayer' type's nodes (i.e., circular** and **double bubble)** for designer's choice.

   After executing the 'type' sublayer, the remaining sublayers (i.e., number and position) of main node 'fuselage' are then executed in the same manner.

3. Likewise, the remaining 5 main nodes are implemented according to the order of precedence, as shown in Figure 5.3.2/3.

3-3. After executing the configuration phase, the system shifts to the **'Design Phase'**. This Design Phase has the layer **'Design Activity'** whose 'select' type is all. The 'select' type 'all' has the same meaning as the 'all' in the layer 'component' of the configuration phase. The layer 'Design Activity' has the following nine(9) main nodes and the number means the order of execution.

1. Parametric Study
2. Fuselage Design

3. Wing Design
4. Engine Design
5. Vertical Tail Design
6. Horizontal Tail Design
7. Undercarriage Design
8. Weight Analysis
9. Cost Analysis

1. Each main node has its sublayer **'Design Template'** whose 'select' type is **'all'**. The 'select' type 'all' has the same meaning as the 'all' in the layer 'component' of the configuration phase and the layer 'design activity' of the design phase.

2. Thus, as shown in the Figure 5.3.2/3, the **sublayer 'design template'** of the main node **'Parametric Study'** has the following five (5) nodes to be executed all, according to the order of precedence. Finishing the execution of all the nodes means the execution of sublayer 'design template' and its main node 'parametric study'.

    1. Payload Range
    2. Take-off Performance
    3. Landing Performance
    4. Cruise Matching
    5. Size Matching

3. Likewise, the remaining eight (8) nodes are executed according to the order of precedence, which means the execution of the layer 'design activity'.

3-4. After finishing the execution of each node in DESAID, a **Backtrack** occurs by either the system itself or designer's wish, as shown in the Figure 5.3.2/3.

1. If the results do not satisfy a certain requirements built in the system, the system checks and shows "where to backtrack" with some alternatives on the screen.

    For example, if the requirements of landing performance were not satisfied with the landing calculation, then the problem is with the landing performance. Thus the system shows, with the "beep" sounds, the recommendation for the designer to check is "Wing loading is not satisfied".

2. If the designer wishes to backtrack whether the results are to his satisfaction or not, the backtrack starts and the system shows "where to backtrack" candidates so that he/she can choose at his/her will.

3. If the backtrack starts, the system deletes all the results after the new start points.

3-5. At any stage during execution, the system can **explain the reasoning process** by showing " How did the system arrive at such results ?" on the screen, depending upon the designer's request.

All the summary of these expressions are shown in the Appendix IV *"Inference Engine"*.

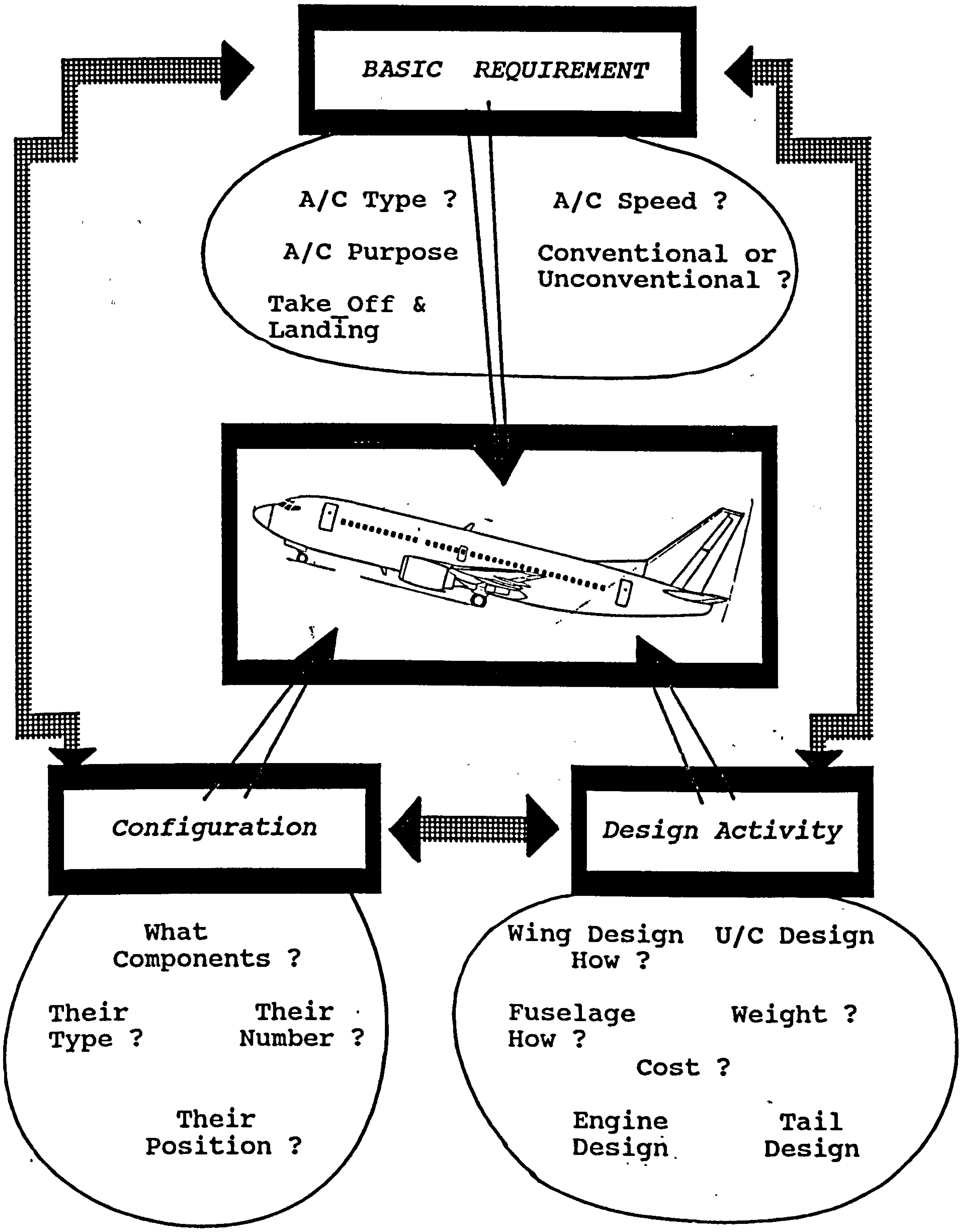


Figure 5.1/1 Complexity of Aircraft Configuration Design

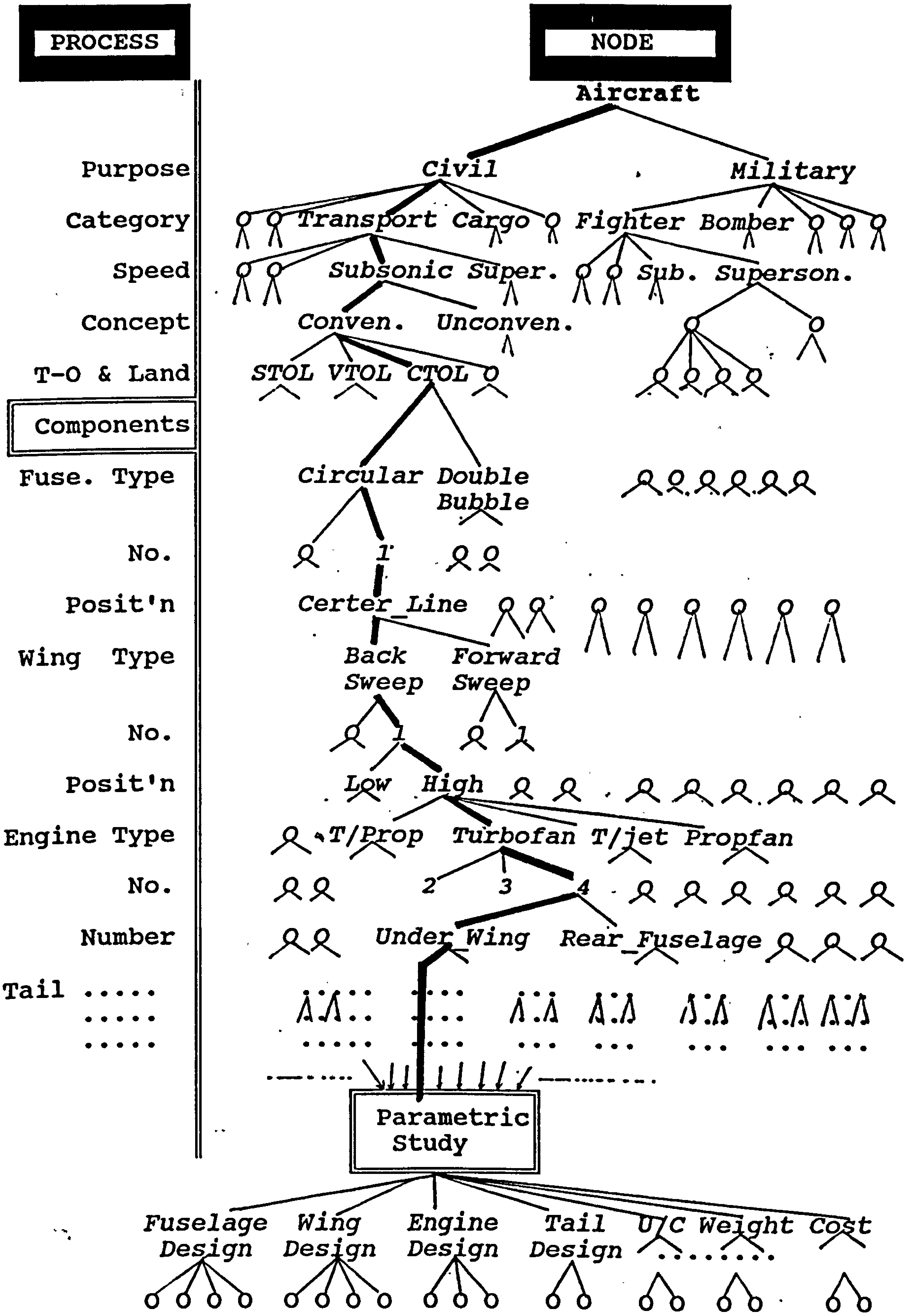


Figure 5.2/1 TREE Structure

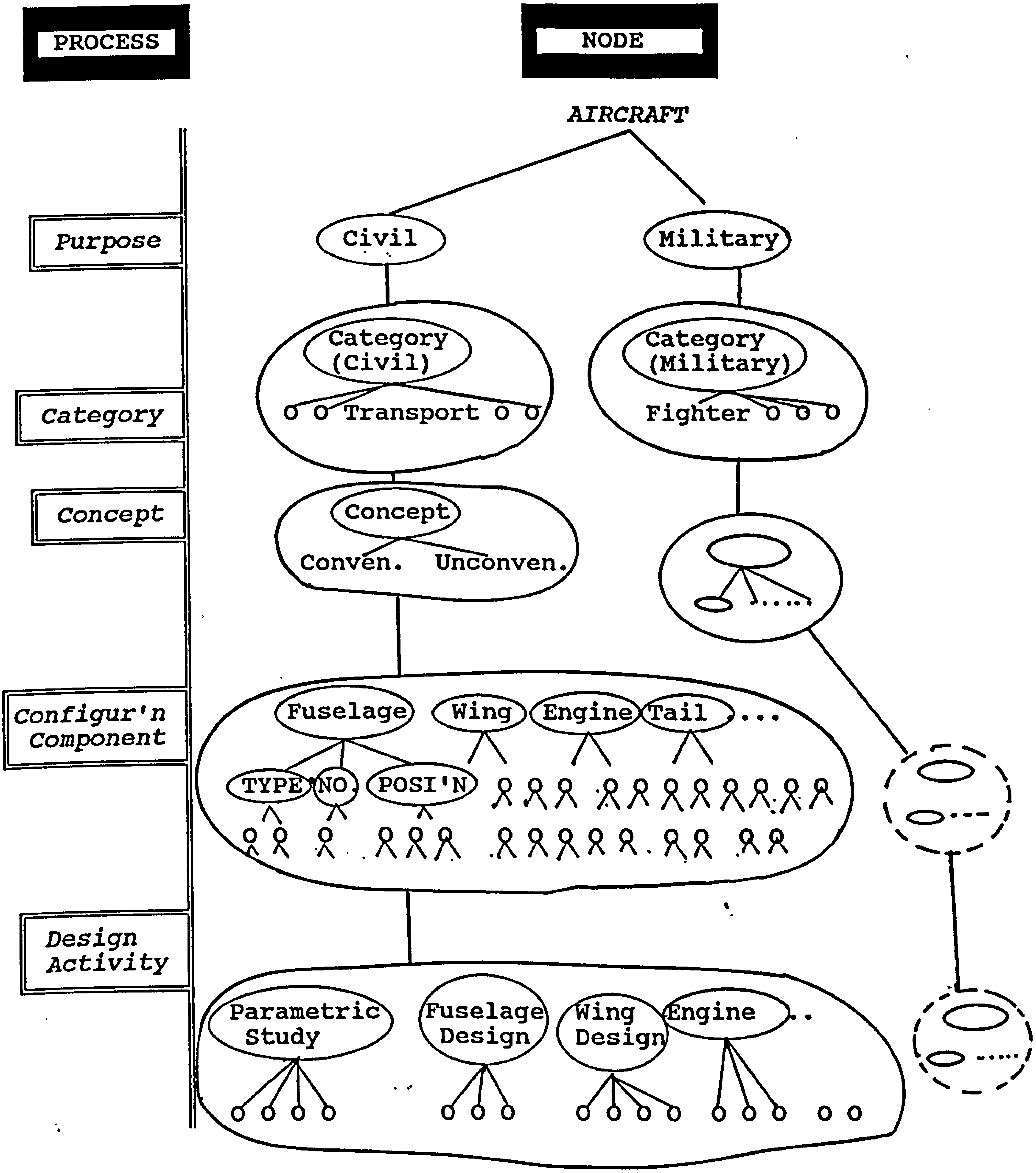


Figure 5.2/2 **Clustered Structure**

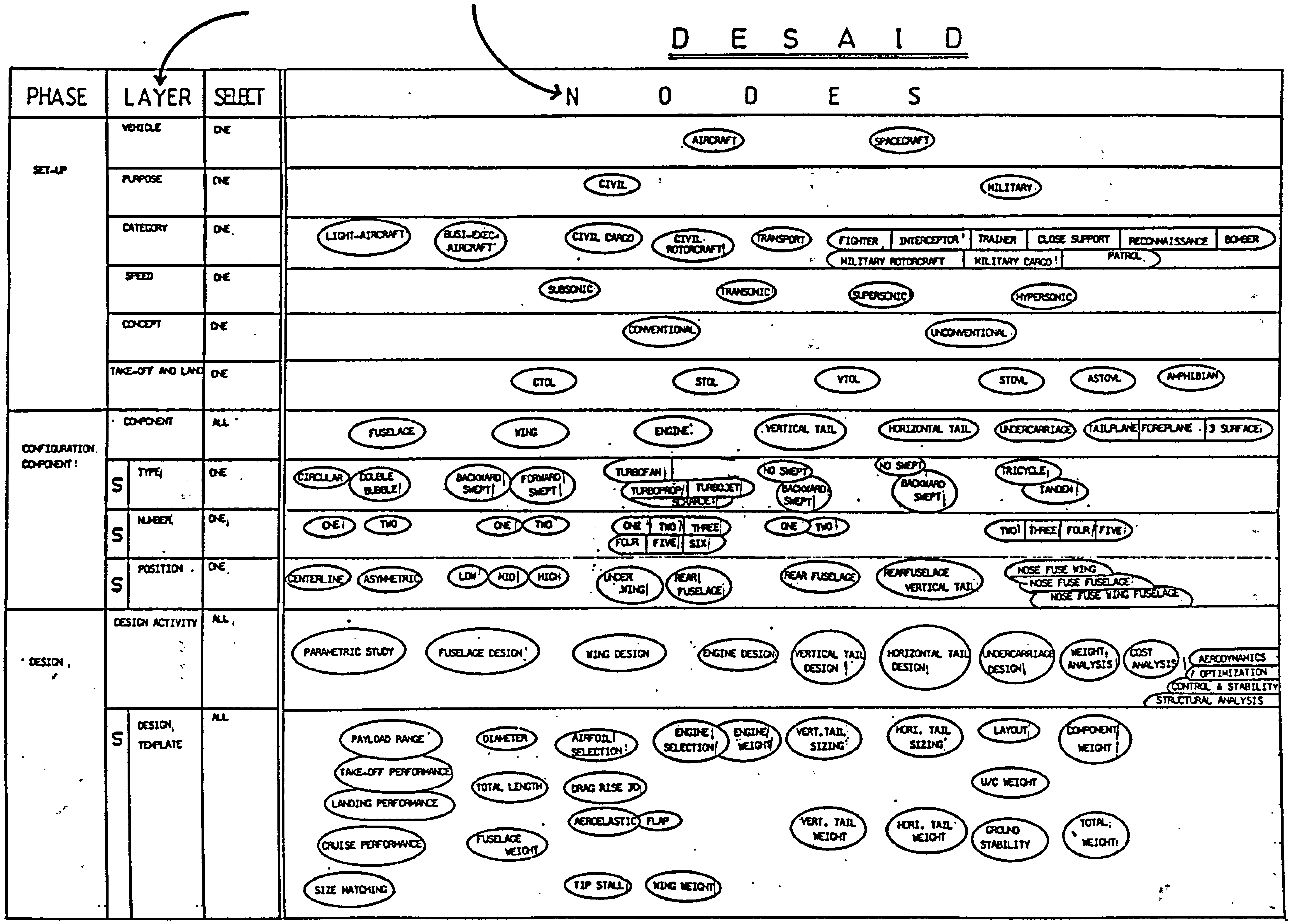


Figure 5.2/3 The Layer and Node Concept

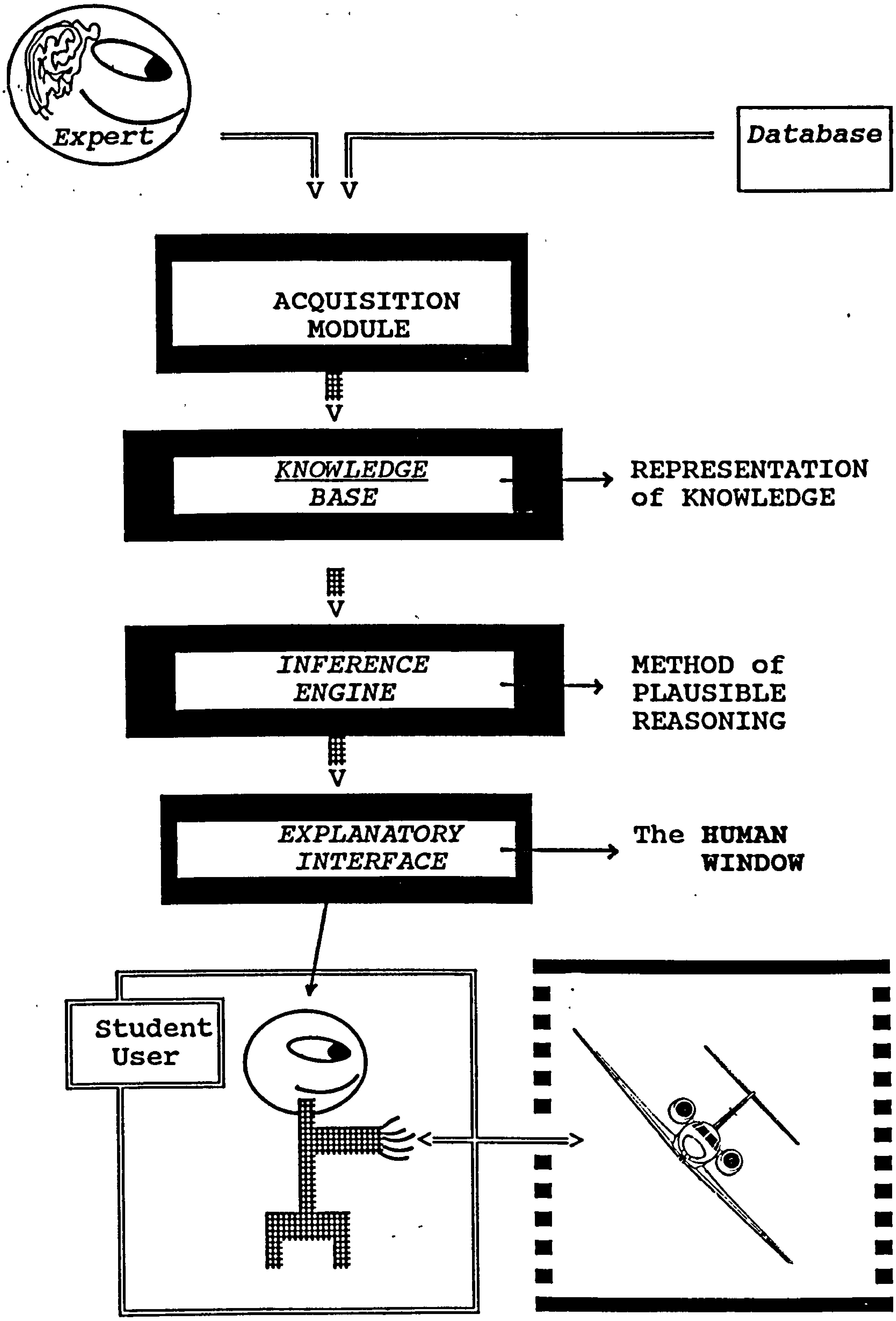


Figure 5.3/1 The Expert System Structure

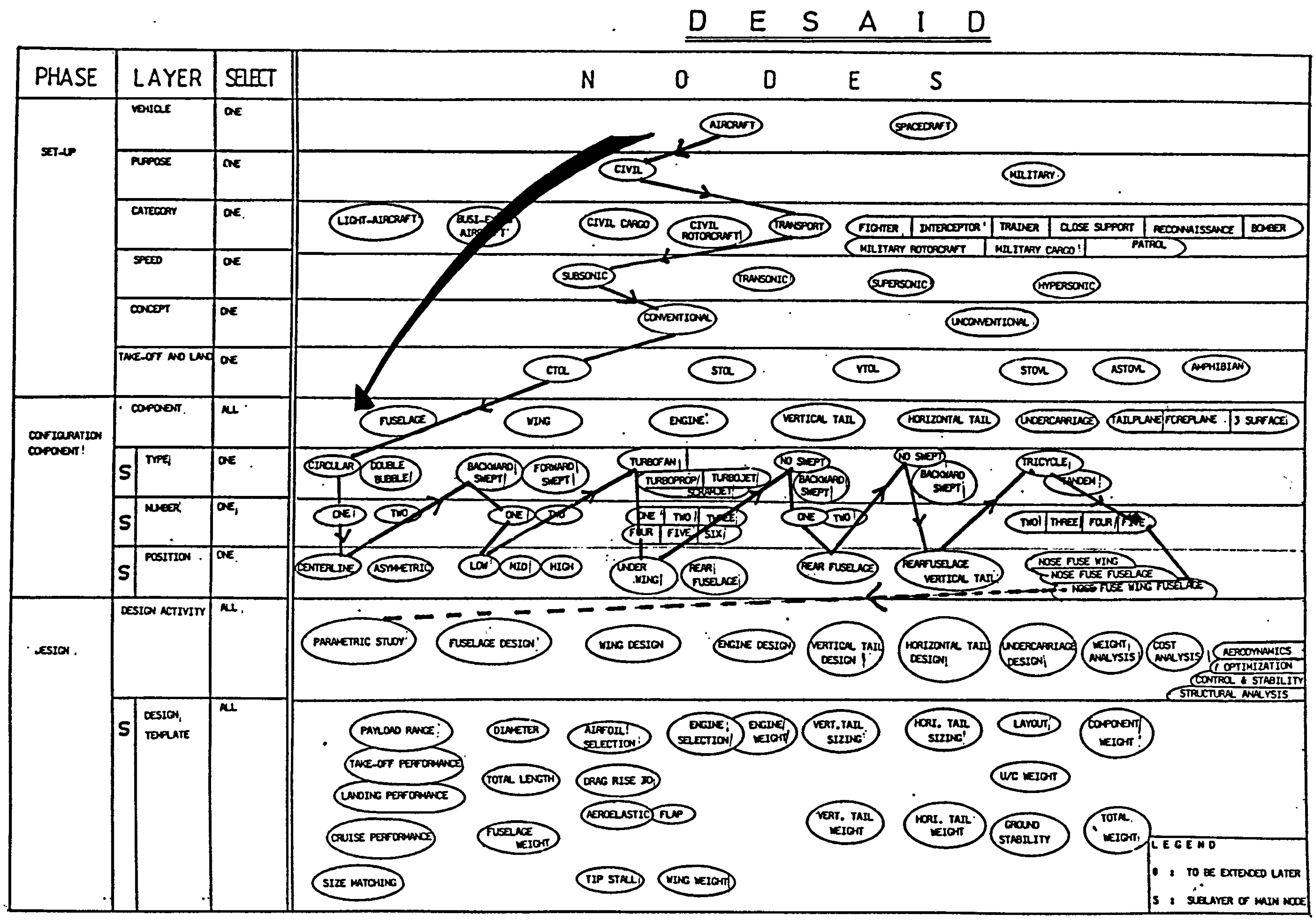


Figure 5.3.2/1 The Forward Chaining Example

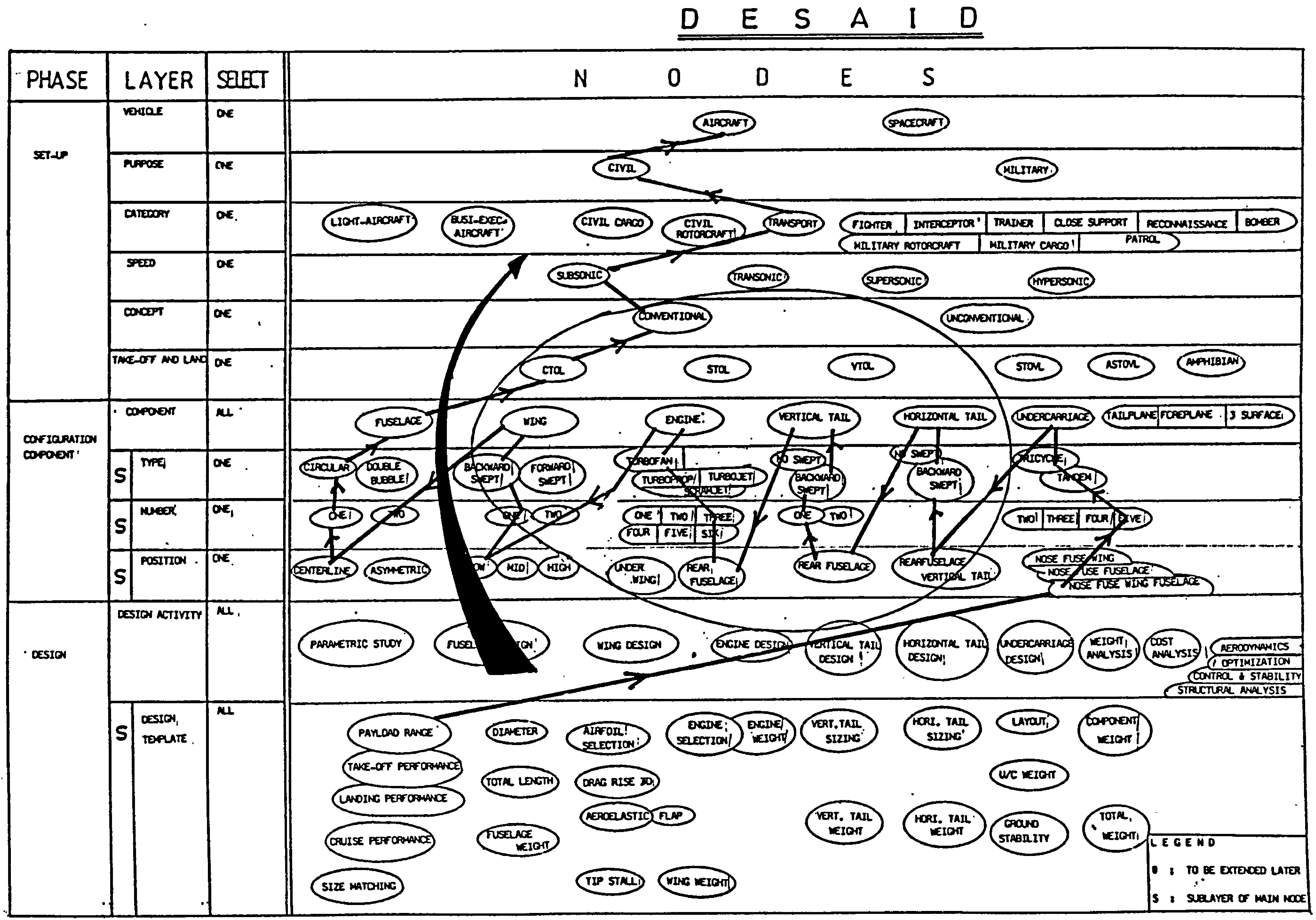


Figure 5.3.2/2 The Backward Chaining Example

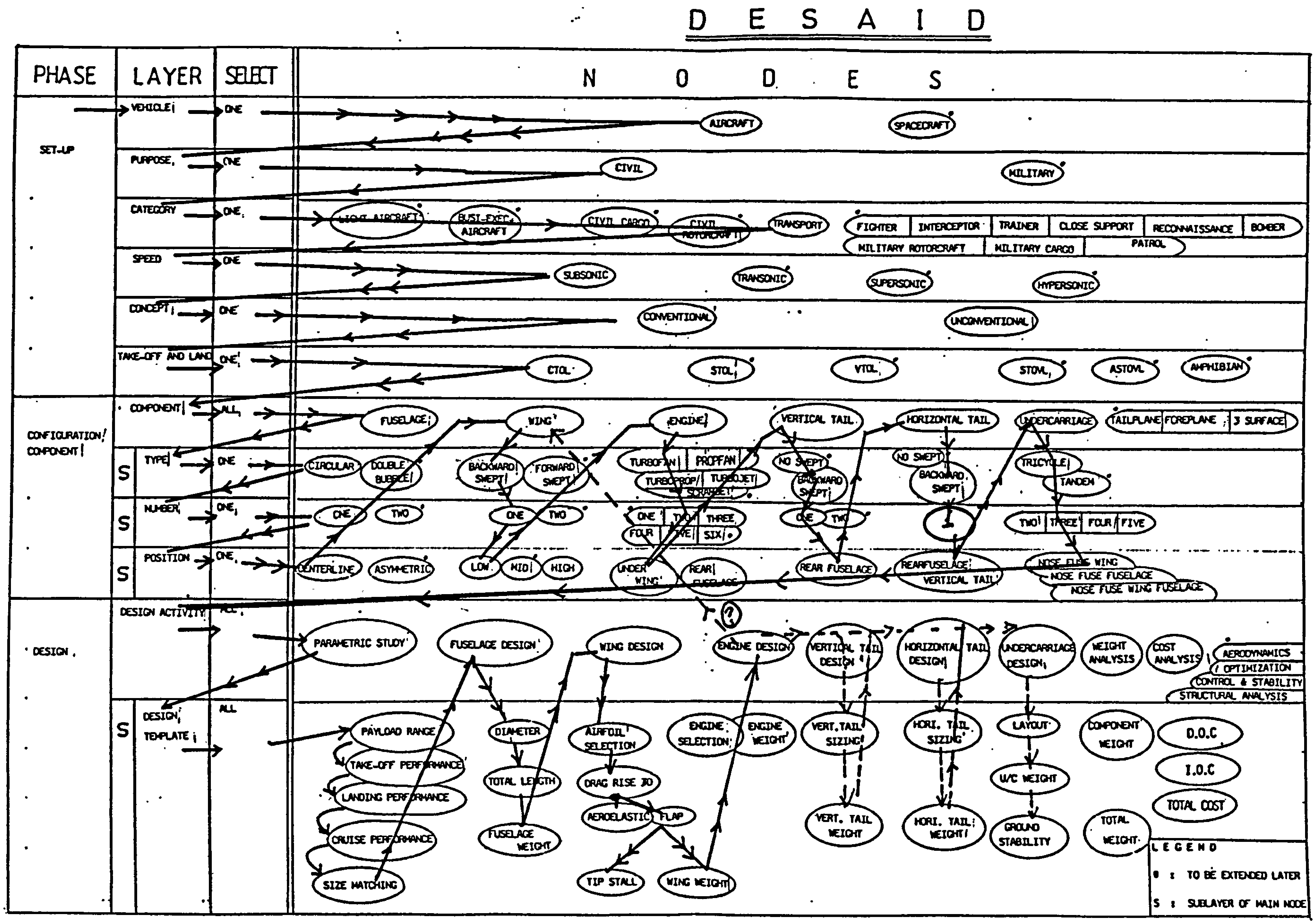


Figure 5.3.2/3 The Search Strategy

CHAPTER 6

DESAID : PROGRAM and OPERATION

" *In turning a formalization of knowledge into a runnable program, one is primarily concerned with the specification of control and the details of information flow. Rules have to be expressed in some executable form under a chosen control regime, while decisions must be made about data structures and the degree of independence between different modules of program.*" -- Buchanan, et al. (1983), [51] [61]

CHAPTER 6 DESAID : PROGRAM and OPERATION

6.1 DESAID : Program

The overall Expert System was developed for aircraft configuration design, called *D E S A I D* (The Development of an Expert System for Aircraft Initial Design). It has the following facilities, as shown in Figure 6.1/1.

1. User Interface : A.I. Computer for Input and Interacting with Designer or User.
2. Knowledge Base : Aircraft Design Knowledge represented in PROLOG Expression.
3. Data Base : RAE Supercritical Airfoil (RAE 95 series Data) and Turbofan Engine data existing at present.
4. Inference Engine : Overall Control Structure which was programmed with PROLOG and described in Appendix IV.
5. Explanation Function : Explains the reason why a certain conclusion has been arrived.

To program this expert system easily with the lessons (Modular Programming) from Chapter 3 incorporated, the program sections were structured in each module. These are explained as follows;

- The Project Name is **VEHICLE.PRJ** and this is the execution file. That means "Typing 'VEHICLE' on the screen runs the program".

Program Module

1. VCLAUSE.PRO : The **Knowledge Base, Fact and Rules** were stored.
2. VCONTROL.PRO : The **Control Mechanism and Inference** Engine were programmed.
3. VMAIN.PRO : The Overall screen handling and menu controlling.
4. VDATA.PRO : The DATA Base.
5. VDBA.PRO : The DATA Handling.
6. VDESAIN.PRO : The Numerical Equation Handling.
7. VCOMMON.PRO : The Screen and Keyboard Handling.
8. VGRAPH.PRO : The Graphic Screen handling.
9. VGR.PRO : The Graphic Result Visualization.
10. VGLOBAL.PRO : Stored the predicate expression used in PROLOG Language.

11. *.TXT : This file is stored in the Disk and accessible during an execution. The Text file contains the explanations of a Layer and a Node.

12. *.MAE : This file is stored in the Disk and accessible during execution. The Meaning and Example file contains the explanation of detailed items during execution.

6.2 Operation Instructions

The system 'DESAID' implements aircraft design by using various windows for an effective user interface.

Prior to the start of design, the input from the specification and design requirements is typed into the system and the input can be updated through menus and screens at any stage of design activity during execution.

The result from execution can be shown in a text or graphical form. For consultation purpose at any stage during execution, the system provides such function as 'Help' and the system informs the user of error occurred, together with the necessary check points.

For easy access to DESAID, all the **files** are provided in a **floppy disk** but only the **execution file** is stored in **Hard Disk** because of its big compilations. The files are as follows;

1. Distribution Disk : ***.TXT and *.MAE** (As explained in the previous section)
2. Hard Disk : The executable image of the system **'VEHICLE.EXE'**

During execution, three types of files can be created as follows;

1. ***.OUT** : stores the results of design step *, together with the file name, directory, date, and time.
2. ***.SAV** : saves the state of design step *, as the designer wishes.
3. ***.REA** : is created as "reasoning file" when the designer wants to know "How can he/she arrive at such conclusion ?".

The **trial implementation in Appendix VI** shows the **design step, input and output, execution of design step, checking design step, file menus, and etc.**

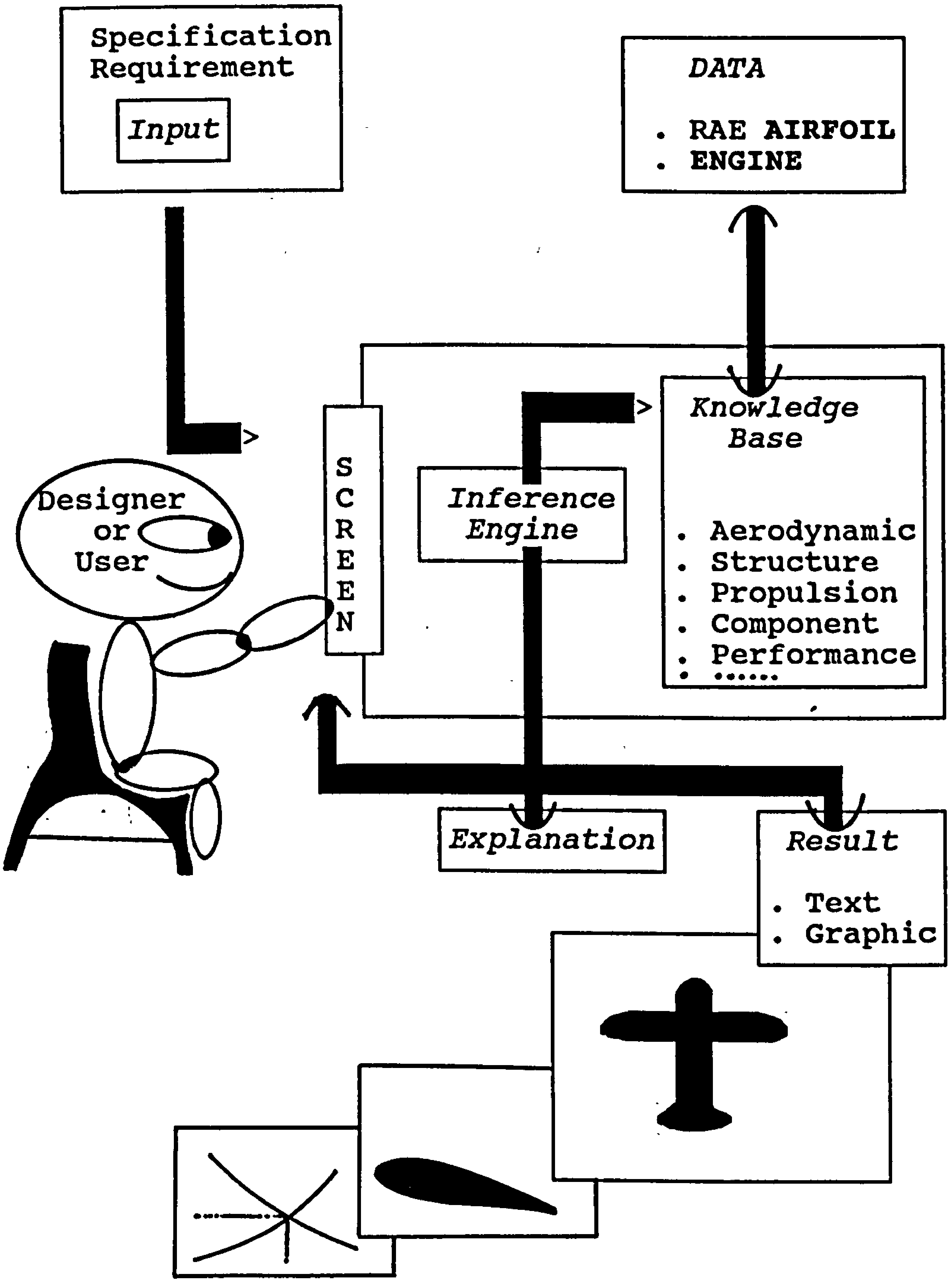


Figure 6.1/1 The Structure of *D E S A I D*

CHAPTER 7

TEST, RESULT, and DISCUSSION

> *"Test cases are useful for other reasons, such as during review of developments to prompt the experts for fuller rules by running them through the system. ------- but who says that expert system implements are less prone to error than the rest of the human race ?"* --- Gilly Furese, Chapter 10, Expert System, 2nd ed. by R. Forsyth, Chapman and Hall. [56]

CHAPTER 7 TEST, RESULT, and DISCUSSION

7. 1 Test

The test was conducted for the current existing aircraft among the conventional, subsonic, and Jet Transports. The following is the list of aircraft which has been experimented with.

Aircraft Company	Aircraft Type	Remarks
1. Airbus	*A-300-600*	
	A-310	
	A-320	
2. Fokker	*F-100*	
3. Soviet Ilyusin	*II-62*	
	II-86	
4. Soviet Tupolev	*Tu-154B/2*	
5. BAe	*BAe-146/100*	
	BAe-146/200	
	BAe-146/300	
6. Boeing	*B-737-200*	
	B-747-200	
	B-757	
	B-767	
7. McDonnell Douglas	*MD-81*	
	DC-10-30	
8. Short Range Airliner	*A-90*	(*)

All of the tests were accomplished, starting with the specification input, parametric study, fuselage, wing design, and engine design (selection). The rests of the design activities already programmed such as tail design,

---

(*) This A-90 aircraft was designed by Dr. J. P. Fielding of Aerospace Vehicle Design Department in Cranfield Institute of Technology. It is 'high_wing', 'two under_wing mounted engine', and 'double_bubble cross section' aircraft. Its specification was described in DES 9000/1. [2]

undercarriage, weight analysis, and cost analysis were excluded due to the computer real memory capacity. That was because the Turbo Prolog software used in this thesis was developed on the basis of a DOS system.

If an Artificial Intelligence software is based on another system such as an "Overlay System" which can extend the real memory during a numerical calculation, it is possible to insert the design activities not tried here into the total system and compile/run accordingly.

7. 2 Result

The test was conducted to validate this system by comparing the test results with the aforementioned aircraft with respect to the **set-up phase, configuration phase, and design phase.**

The test results of A-90 aircraft are shown as an example with respect to the three phases, as shown in the Figure 7.2/1. After having checked the set-up and configuration phases of the aforementioned aircraft, the numeric values were compared between the existing aircraft(*) and this system's test results.

7.2.1 Test of the **Set-up Phase : Figure 7.2/1**

Phase	**Layer**	**Node's Result**	**Input by user**	**Input by system**
Set-up	Vehicle	**aircraft**	o	
	Purpose	**civil**	o	
	Category	**transport**	o	
	Speed	**subsonic**		o
	Concept	**conventional**	o	
	Take-Off & Land	CTOL		o

7.2.2 Test of the **Configuration Phase : Figure 7.2/1**

(*) The specifications of the existing aircraft were collected from *Jane's All the World's Aircraft*. [9]

Layer	Main Node	Sublayer's Node Result	Input by user	Input by system
Component	Fuselage	double_bubble	O	
		one		O
		center_line		O
	Wing	backward_swept		O
		one		O
		high	O	
	Engine	turbofan		O
		two_engine	O	
		under_wing mount		O
	Vertical Tail	backward_swept		O
		one		O
		rear_fuselage_mount		O
	Horizontal Tail	backward_swept		O
		one		O
		vertical_tail_mount		O
	Undercarriage	retractable/tricycle		O
		five		O
		nose_fuse_fuselage		O

7.2.3 Test of the Design Phase

The Test Results showed good agreement when with modern medium to large transports. However, a difference could be encountered in case of both the small commuter and Soviet Aircraft.

The small commuter's case may be attributed to the fact that this research focused on high subsonic wide body transports and the soviet's case seems to be due to the use of an aircraft design method different from that of western aircraft manufacturers. Another discrepancies in engine and thrust loading were due to the lack of realistic engine data available at the moment.

Followings are the summary of test and the differences were more or less about 10 %.

Aircraft	Area	Test Result	Real Value	Error
1. A-310	W_G(lb)	342865	330695	3.7 %
	W_E(lb)	181474	169321	7.2 %
	T_O(lb)	98074	96000	1.2 %
	Dia(ft)	21	18.5	15.0 %
2. A-300 600	W_G(lb)	349602	363765	3.9 %
	W_E(lb)	164536	175884	6.5 %
	S_W(ft^2)	2630	2798	6.0 %
	Dia(ft)	18	18.5	3.2 %
	L_{fuse}(ft)	172	175	1.7 %
	W_G / S_W	132	130	1.5 %
3. A-320	W_G(lb)	136369	145505	6.3 %
	W_E(lb)	70813	82895	14.5 %
	S_W(ft^2)	1304	1313	0.7 %
	Dia(ft)	13	13.3	3.0 %
	L_{fuse}(ft)	120	123	2.1 %
	W_G / S_W	110	111	0.5 %
4. F-100	W_G(lb)	82019	91490	10.4 %
	W_E(lb)	43314	51147	15.3 %
	S_W(ft^2)	849	1006	15.6 %
	T_O(ft)	23367	27000	13.4 %
	Dia(ft)	10	10.8	8.0 %
	L_{fuse}(ft)	97	107	8.2 %
	$Span_W$	84	92	8.1 %
5. Il-86	W_G(lb)	386163	418875	7.8 %
	S_W(ft^2)	3099	3444	10.1 %
	T_O(lb)	111248	114640	3.0 %
	Dia(ft)	19	19.9	5.0 %
	L_{fuse}(ft)	185	184	0.1 %
	$Span_W$	149	157	4.8 %
	W_G / S_W	124	122	2.5 %
6. Tu-154	W_G(lb)	158580	185188	14.3 %
	W_E(lb)	85542	95900	10.9 %
	T_O(lb)	46903	69450	32.4 %
	Dia(ft)	10	12.5	24.0 %
7. BAe-146-100	W_G(lb)	73345	84000	12.7 %
	W_E(lb)	38344	49000	21.7 %
	S_W(ft^2)	722	832	13.2 %
	Dia(ft)	11	11.8	7.6 %
	L_{fuse}(ft)	80	86.5	7.0 %
8. BAe-146-200	W_G(lb)	91118	93000	2.0 %
	W_E(lb)	47108	50400	6.5 %
	S_W(ft^2)	1037	832	24.6 %
	Dia(ft)	13	11.8	10.9 %

	L_{fuse}(ft)	105	94	11.6 %
	T_O / W_G	0.274	0.299	8.4 %
9. BAe-146-300	W_G(lb)	104061	104000	0.1 %
	W_E(lb)	54046	57100	5.3 %
	T_O(lb)	28776	30000	4.0 %
	L_{fuse}(ft)	109	104	4.8 %
	W_G / S_W	113	125	9.6 %
	T_O / W_G	0.276	0.268	2.9 %
10. B-737 -200	W_G(lb)	112311	115500	2.8 %
	W_E(lb)	57557	60210	4.4 %
	S_W(ft^2)	1133	1098	3.2 %
	T_O(lb)	30316	29615	2.4 %
	L_{fuse}(ft)	98	97	1.7 %
	W_G / S_W	99	118	15.3 %
	T_O / W_G	0.270	0.277	2.9 %
11. B-747 -200B	W_G(lb)	818935	820000	0.1 %
	W_E(lb)	366921	375000	2.2 %
	W_f(lb)	323050	324480	0.4 %
	S_W(ft^2)	6187	5500	12.5 %
	T_O(lb)	170451	210000	18.8 %
	Dia(ft)	21.7	20.5	6.1 %
	L_{fuse}(ft)	238.7	225	5.8 %
	$Span_W$(ft)	207.5	195.8	6.0 %
	W_G / S_W	132	149	11.4 %
	T_O / W_G	0.208	0.256	18.7 %
12. B-757	W_G(lb)	246700	240000	2.8 %
	W_E(lb)	122720	126250	2.8 %
	S_W(ft^2)	2297	1994	15.2 %
	T_O(lb)	63034	74800	15.7 %
	Dia(ft)	13	12	8.3 %
	L_{fuse}(ft)	165	155	6.4 %
	$Span_W$(ft)	133.6	124	7.2 %
	W_G / S_W	107	110	2.7 %
13. B-767	W_G(lb)	339050	345000	1.7 %
	W_E(lb)	171666	180600	4.9 %
	S_W(ft^2)	3499	3050	14.7 %
	Dia(ft)	17.7	16.5	6.7 %
	L_{fuse}(ft)	170	176	3.4 %
	$Span_W$(ft)	166	156	6.3 %
14. MD-81	W_G(lb)	136919	140000	2.2 %
	W_E(lb)	67147	73157	8.2 %
	S_W(ft^2)	1305	1270	2.8 %
	T_O(lb)	34061	37100	8.2 %
	Dia(ft)	10.2	10.5	2.8 %
	L_{fuse}(ft)	137	135.5	1.3 %
	$Span_W$(ft)	112	107.8	3.9 %
	W_G / S_W	105	109	3.7 %
	T_O / W_G	0.249	0.267	7.1 %
15. DC-10 -30	W_G(lb)	585369	580000	1.0 %
	W_E(lb)	279963	267197	4.7 %
	S_W(ft^2)	4373	3958	10.5 %

	Dia(ft)	19.9	19.5	2.1 %
	L_{fuse}(ft)	191.7	182	5.3 %
	Span$_W$(ft)	175	165.5	5.7 %
	W_G / S_W	133	144.5	7.9 %
16. A-90	W_G(lb)	428640	465340	8.6 %
	W_E(lb)	213562	213442	0.1 %
	S_W(ft^2)	3416	3890	13.8 %
	Dia(ft)	22	23.5	6.8 %
	L_{fuse}(ft)	198	195	1.5 %
	Span$_W$(ft)	175	187	6.8 %

7.3 Discussion

Even if the system results were fitted well with the real values of existing aircraft, there are some things to be desired for the further development and system enlargement as follows;

First, in the language option, another Artificial Intelligence language is recommended to be used so that the limit of the DOS system Memory 640K can be surmounted.

Second, for the further enlargement and execution of numerical calculation in such areas as optimization, aerodynamic analysis, and structural analysis, a more powerful language is preferable. Thus the author would like to suggest the C^{++} - objected oriented language, in consideration of the fact that the Concept of "LAYER and NODE" used herein is very close to that of an object oriented approach.

Third, in case of an enlargement, such design activities as stability & control, aerodynamic analysis, structural analysis, etc., should be incorporated for the total integration of aircraft design at the initial design stage. In addition, there is a need to rubberize the actual engine size, it is essential to make use of the actual engine data available from the engine manufacturers, even though it is usually difficult to get the realistic engine data in the Institute Research environment.

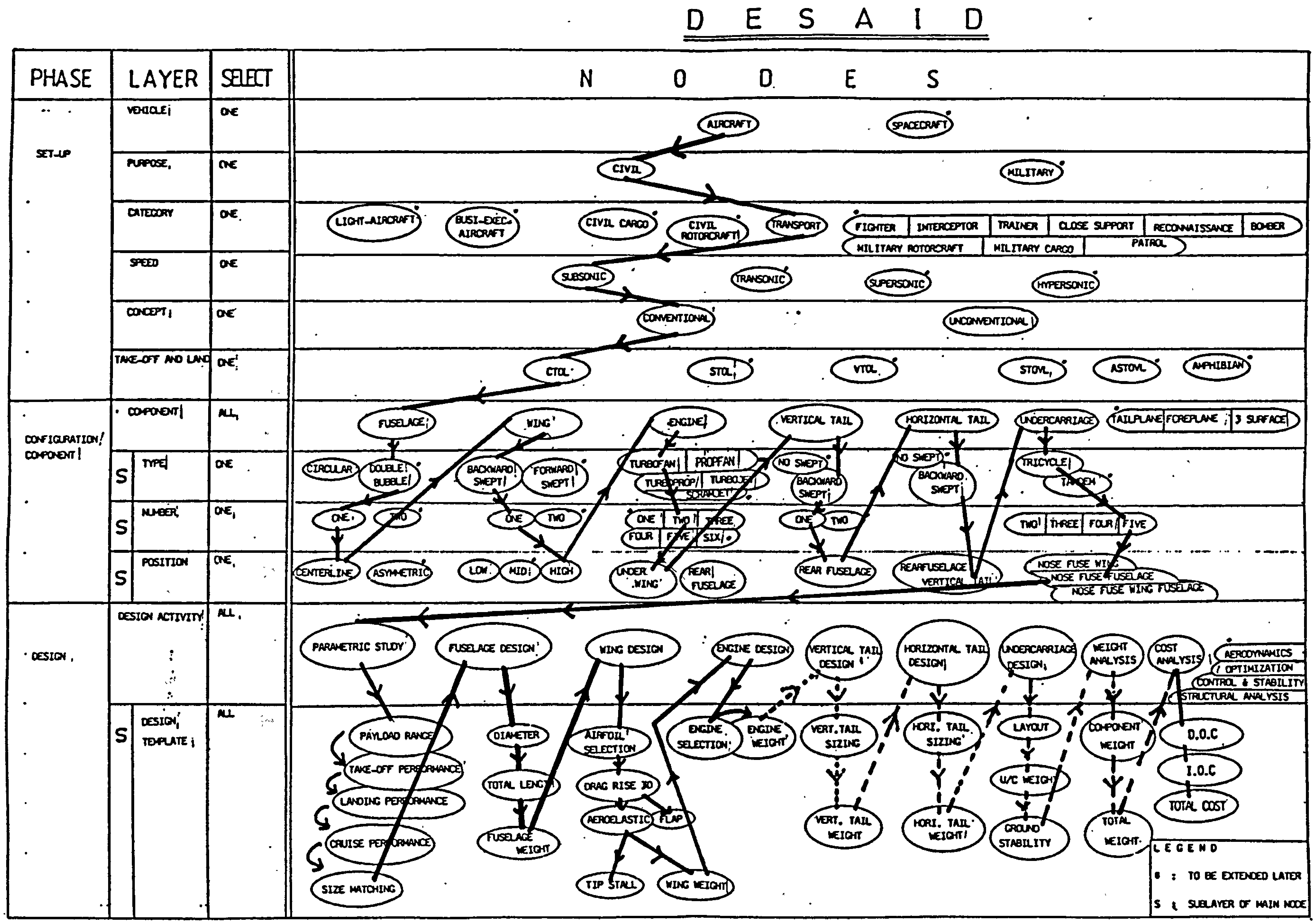


Figure 7.2/1 Test Example of A-90 Aircraft

REFERENCES

1. ALSINA, J., Development of an Aircraft Design Expert System, Ph.D Thesis, 1988.
2. HOWE, D., and FIELDING, J.,P., Lecture Note on Aircraft Design, Cranfield Institute of Technology, 1988.
3. TORENBEEK, E., Synthesis of Subsonic Airplane Design, Delft University Press, 1981.
4. CORNING, G., Subsonic and Supersonic, CTOL and VTOL Airplane Design, 1976.
5. NICOLAI, L., M., Fundamentals of Aircraft Design, MET S., Inc., 1975.
6. ROSKAM, J., Airplane Design, Volume 1, 2, 3, 4, 5, 6, 7, 8, Roskam Aviation/Engineering Company, 1985.
7. LOFTIN, K., L., Subsonic Aircraft Performance, Design, and Size, NASA RP - 1060, AUG., 1980.
8. CIPOLLONE, G.A.F.G., Development of an Expert System for Fuselage Design, M.Sc. Thesis, Cranfield Institute of Technology, Bedford, England, 1986.
9. TAYLOR, JOHN, W., R., ed. : Jane's All the World's Aircraft, Jane's, London, England, UK, 1986-1990.
10. CURRY, N., S., Landing Gear Design Handbook, Lockheed - Georgia Company, 1982.
11. FAA, FAR Part 25 : Airworthiness Standard : Transport Category Airplanes, U.S.A., FAA.
12. FIELDING, J., P., Lecture Note on Aircraft Design, Cranfield Institute of Technology, 1988.

13. FULKER, J., L., RAE Report and Memoranda No. 3820 - Aerodynamic Data for 3 Supercritical Airfoils, RAE 9515, 9530, 9550, Royal Airforce Establishment, 1975.

14. SLIWA, S. M., and ARBUCKLE, P. D., OPDOT : A Computer Program for the Optimum Preliminary Design of a Transport Airplane. NASA / TM -81875, September, 1981.

15. KUCHERMANN, D., The Aerodynamic Design of Aircraft, Pergamon Press, 1978.

16. McRAE, D., M., The Aerodynamic Development of the Wing of the A300B, Journal of Royal Aeronautical Society, July, 1973.

17. McCORMICK, B., W., Aerodynamics, Aeronautics, and Flight Mechanics, John Wiley & Sons, 1979.

18. RAYMER, D.,P., Aircraft Design : A Conceptual Approach, AIAA Education Series, AIAA, 1989.

19. PASARIBU, H., M., A Computer Augmented Procedure for TRANSPORT AIRCRAFT FUSELAGE DESIGN SYNTHESIS, M. Sc. Thesis, 1988.

20. PRATT & WHITNEY, The Aircraft Gas Turbine and its Operations, United Technology Corporation, 1988.

21. ROLLS-ROYCE, The Jet Engines, Rolls-Royce, LTD, 1966.

22. STINGTON, D., The Design of the Aeroplane, BSP, 1983.

23. STINGTON, D., The Anatomy of the Aeroplane, BSP, 1983.

24. TORENBEEK, E., Analytical Method for Computing Turbo - Engine Performance at Design - Off Design Conditions, Delft University, 1973.

25. BOWERS, P. M., Unconventional Aircraft, TAB, 1984.

26. LANGE, Roy H., Review of Unconventional Aircraft Design Concepts, Volume 25, No. 5, Journal of Aircraft, May 1988.

27. DRESSE, M., L., Twin / Four - Engined Jets, A comparative study, Institut du Transport, Aerien - Paris, 1986.

28. LEE, V., A., Computerized Aircraft Synthesis, Journal of Aircraft Vol. 4, No. 5, Sep-Oct, 1967.

29. WALLACE, R., E., A computerized system for the preliminary design of commercial Airplanes, AIAA 4th Aircraft Design Flight Test and Operation meeting, Los Angeles, California / August 7 -9, 1972.

30. PIGGOT, B., A., M., / TAYLOR, B., E., Application of Numerical Optimization to the preliminary design of a transport aircraft, RAE Technical Report 71074, April, 1971.

31. KIRKPATRIC, D., L., I. / PECKHAM, D., H., Multivariate Analysis Applied to Aircraft Optimization, RAE Technical Memorandum Aero 1448, Sep. 1972.

32. HAGUE, D., GASP - General Aviation Synthesis Program, Part I/II, NASA - CR 152303, Jan., 1978.

33. SLIWA, S., M., Use of Constrained Optimization in the Conceptual Design of a Medium Range subsonic Transport, NASA Technical Paper 1762, 1980.

34. HABERLAND, C., THORBECK, J., FENSKE, W., A Computer Augmented Procedure for Commercial Aircraft Preliminary design and optimization, ICAS - 84 - 4.8.1.

35. SNEPP, D., K., POMEROY, R., C., A Geometry System for Aerodynamic Design, AIAA - 87 - 2902, Sep 1987.

36. KROO, I., TAKAI, M., A Quasi - Procedural , Knowledge - Based System for Aircraft Design, AIAA - 88 - 4428, Sep., 1988.

37. ANDERSON, B., L., First Step Toward Integrating the Design Process, AIAA - 88 - 4403, Sep., 1988.

38. MARINOPOLOUS, S., JACKSON, D., SHUPE, J. and MISTREE, F., University of Houston, AIAA - 87 - 2965, September, 1987.

39. SALGUERO, D., E., A Design Synthesis Program for Business Jet Aircraft.

40. PASARIBU, H. M., Survey of Existing Aircraft Design Computer Programs, unpublished, C.I.T, Oct, 1989.

41. JENKINSON, L., R. and SIMOS, D., Optimization of the Conceptual Design and Mission Profiles of Short - Haul Aircraft, Journal of Aircraft, Vol. 25, No. 7, July, 1988.

42. BARGETTO, R., MAZZETTI, B. and GARBDINO, G., Aircraft Configuration Analysis / Synthesis Expert System : A New Approach to Preliminary Sizing of Combat Aircraft, ICAS - 88 - 1.11.2, 1988.

43. ROSKAM, J., MALAEK, S. M., and ANEMAAT, W., AAA(Advanced Aircraft Analysis) : A user-friendly approach to preliminary aircraft design, ICAS-90-2.10.2, ICAS, Sep., 1990.

44. IPAD, Integrated Programs for Aerospace Vehicle Design, NACR - 2983, NASA, 1979.

45. BILL, S., and BUSICK, J., D., Future Flight, TAB, 1987.

46. ANDERSON, Jr., J., D., Introduction to Flight. McGraw - Hill, 1989.

47. CONSOLI, R., D. and SOBIESKI, J., S., "Application of Advanced Multidisciplinary Analysis and Optimization Method to Vehicle Design Synthesis, ICAS-90-2.3.4, ICAS, Sep., 1990.

48. HOLTZ, M., Mastering Microsoft Word on the IBM PC, 4th ed., SYBEX, 1989.

49. COYNE, R., D., Rosenman, M., A., RADFORD, A, D., Knowledge Based Design System, Addison - Wesley, 1990.

50. QUINLAN, J., R., Applications of Expert Systems, Addison - Wesley, 1989.

51. JACKSON, P., Introduction to Expert System, Addison-Wesley, 1986.

52. RICH, E., Artificial Intelligence, Mcgraw-Hill, 1985.

53. CLOCKSIN, W., F., and MELLISH, C., S., Programming in PROLOG, Springer-Verlag, 1981.

54. ALSINA, J., ANINOSIS, A., EROL, N., FIELDING, J., P., and MORRIS, A., J., Progress Toward a Commercial Aircraft Design Expert System, presented at DGLR Conference, Berlin, W., Germany, Oct., 1987.

55. ARNOLD, W., R., and BOWIE, J., S., Artificial Intelligence, A personal Common Journey, Prentice-Hall, Inc., Englewood-Cliffs, 1986.

56. FORTHSYTH, R., Expert Systems : Principles and Case Studies, Chapman and Hall, 1984.

57. BARR, A., and FEIGENBAUM, E., A., Eds.(1981), The Handbook of Artificial Intelligence, Vol. 1/2, Calif., Morgan Kaufmann, 1981.

58. CHARNIAK, E., and McDERMOTT, D., Introduction to Artificial Intelligence, Addison-Wesley, 1985.

59. SYKES, J., B., 6th Ed, The Concise Oxford Dictionary of Current English, Oxford University Press.

60. HILGARD, E., R., & BOWER, G., H., Theories of Learning, Prentice-Hall, Englewood-Cliffs, N. J., 1975.

61. WATERMAN, D., A., A Guide to Expert System, Addison-Wesley, 1986.

62. RYCHENER, M., D., Expert Systems for Engineering Design, Academic Press, Inc., 1988.

63. GERO, J., S., Knowledge Engineering in Computer Aided Design, North-Holland, 1985.

64. KUSIAK, A., Expert Systems, Strategies and Solutions in Manufacturing Design and Planning, S.M.E., 1988.

65. MARCELLUS, D., H., Expert Systems Programming in Turbo- PROLOG, Prentice-Hall, Englewood-Cliffs, 1989.

66. HEPBURN, P., H., Further Programming in PROLOG, Ellis Horwood Ltd., 1987.

67. GARAVAGLIA, PROLOG, Programming Techniques and Applications, Harper & Row Publisher, N.Y., 1987.

68. MONK, A., Fundamentals of Human-Computer Interaction, Academic Press, 1989.

69. MELLISH, C., and HALLAM, J., Advances in Artificial Intelligence, John Wiley & Sons, 1987.

70. NIEGEL, F., How machine think, A General Introduction to Artificial Intelligence, illustrated in PROLOG, John Wiley & Sons, 1987.

71. WALKER, A., A logical approach to Expert Systems and Natural Languages, 1987.

72. WINSTON, P., H., Artificial Intelligence, 2nd ed., Addison - Wesley, 1984.

73. BRAKTO, I., PROLOG Programming in Artificial Intelligence, Addison-Wesley, 1986.

74. BUNDY, A., The Computer Modelling of Mathematical Reasoning, Academic Press, 1983.

75. WOLVERTON, V., Running MS-DOS (2nd Ed.), Microsoft Press, 1984.

76. FOX, J., Expert Systems, The State of the Art, Pergamon, 1984.

77. BOLAND INTERNATIONAL, The Turbo PROLOG, Natural Language of Artificial Intelligence, 1986.

78. HALINA, MORTIMER, The Logic of Induction, Ellis Horwood, 1988.

79. HAYES-ROTH, F., WATERMAN, D., A., and LENAT, D., Building Expert Systems, Addison-Wesley, 1983.

80. HU, DAVID, Programmer's Reference Guide to EXPERT Systems, Horward W. & Sams Company, 1987.

81. KONONENKO, I., and LAURAC, N., PROLOG through Examples, A Practical Programming Guide.

82. WEISKAMP, K., and HENGLE, T., Artificial Intelligence Programming with Turbo PROLOG, John Wiley & Sons, 1988.

83. KOWALSKI, R., Logic for Problem Solving, North-Holland, 1979.

84. MAHER, M., L., and FENVES., S., J., HI-RISE : An Expert System for the Preliminary Design of Hi-Rise Buildings, Carnegie-Mellon University, 1984.

85. MORONE, J., J., Experiencing Artificial Intelligence, TAB, 1987.

86. PHILLIPS, C., Smalltalk V, Personal Computer Word, Nov., 1986.

87. RAUCH - HINDIN, W., B., Artificial Intelligence - Business, Science, and Industry, Vol. 1, Prentice - Hall, 1986.

APPENDIX I

DETAILED ANALYSIS OF AIRCRAFT DESIGN KNOWLEDGE

-- CONTENTS --

I.1	Parametric Study
I.2	Wing Design Analysis
I.3	Fuselage Design Analysis
I.4	Engine Design (Selection)
I.5	Tail Design Analysis
I.6	Undercarriage
I.7	Weight Analysis
I.8	Cost Analysis

I.1 Parametric Study

I.1.1 Specification Study

1. Number of Passengers

Reference [4] recommended the following appropriate seat-abreast seating arrangement depending upon the number of passengers.

PAX	40-65	66-130	131-260	261-420	421-500	500-
SEAT ABREAST	4	5	6	8	9	?

Note) 1. The PAX is an abbreviation of PASSENGER.

2. The Numbers of PAX / SEAT ABREAST can be somewhat varied according to the designers' choice.

3. All the seats are assumed to be for normal / economy class, as the first class can be arranged by varying slightly the seat arrangements according to the designer's needs.

2. Range Classification

(Unit : n.m)

Range item	short	Mid	Long
Nautical Miles	500 - 1999	2000 - 3000	3000 - ?
Reserve Fuel	400 - 600	400 - 600	400 - 600

Climb/Descent	100 - 200	100 - 200	100 - 200

Note) 1. These Data are cited from the reference [7].

2. Reference [7] did not split the reserve fuel and climb/descent but the author regarded it reasonable to divide them approximately.

3. It requires a fairly detailed calculation to determine the amount of fuel reserve exactly. So it is suggested to add an increment of 400 to 600 miles to the design range for roughly estimating an average reserve fuel requirement. Sometimes the fuel reserves are normally specified in the mission specification or the FAR 25 which regulate the operation of passenger transport. [6] [7]

3. Number of Crews

Range Crew	Short-Range	Mid-Range	Long-Range
Pilots	2	2	2
Cabin Crew	PAX/40	PAX/35	PAX/30

Note) 1. The Above data are collected from reference [3].

2. The trend in the number of pilots, considering the state of the art technology of cockpit control layout , assumes 2 pilots.

4. Payload (Unit Weight)

- The Figures are weights per passenger, pilot, cabin crew, or items' unit.

(Unit : lb, lb/cu. ft, %)

Range ITEM	Short	Mid	Long	Remark

Passenger	175	175	175	per pax
Cabin Crew	175	175	175	per crew
Baggage	40	40	40	pax/crew
Cargo	50	50	50	per pax
Cargo Density (lb/cu.ft)	10	10	10	
Cargo Efficiency	85 %	85 %	85 %	

5. Altitude

Altitude, measured from the mean sea level, is different from the Height which is measured from the ground level of some locations.

A Normal cruise altitude is 30,000 up to 45,000 ft and can be slightly varied according to the designer. However, The regional passenger liner's altitude can be assumed to be 25,000 ft. In this research, due to the insufficient engine data in the cruising condition, the range is assumed to fall within 30,000 to 40,000 ft.

6. Cruising Mach Number

The Normal subsonic range falls within the Mach No. 0.7 to Mach No. 0.99. Of course, a Mach number over 0.9 can be regarded to be in a transonic region.

7. Take-off and Landing field Length

The Take-off and Landing field length fall within 5000 to 10,000 ft and the maximum 5,000 to 6,000 ft respectively.

So, the designer can select field length depending upon the airport conditions.

8. Engine By-pass ratio (Turbofan Engine)

The By-pass ratio widely used in the turbofan engines is currently 4.5 to 6.

The 'state of the art' technology of Turbofan engine can extend the ratio range near 10 or above. This can have influences on reducing the specific fuel consumption and extending the flight range. However, the effects from high By-pass ratio over 6.0 can not be incorporated due to the unavailability of engine data.

9. Others

- The Pressurization maintains generally that of altitude 5000 ft at 35,000 ft.
- The Direct climb to a cruising altitude is assumed.
- The requirements of the Federal Aviation Regulations are incorporated for the certification purpose.
- A Mission Profile [6] was shown in the Figure I.1.1/1.

I.1.2 Input and Output Data

In addition to the requirements stipulated in the specification, it is necessary to select , by 'rule of thumb', the reference aircraft which is similar to the aircraft to be designed with respect to the type and size. However, with respect to the aircraft to be designed, the designer assumes the required parameters such as

1. Aspect ratio (typically 5 to 10)
2. Fineness Ratio (" 6 to 12)
3. The Ratio of Fuselage Section Area versus Wing Area : Normal Range 0.03 to 0.15
4. The Size Ratio of Aircraft to be designed versus the Reference Aircraft : it depends upon the designer.

The ratio of drag coefficient to the reference aircraft, $C_{D,0}$ can be expressed as

$$\frac{C_{D,0}}{C_{D,0\ ref}} = \left[\frac{R.N_{ref}}{R.N}\right]^{1/6} = \left[\frac{1.63*10^{4}}{V_c * C_r}\right]^{1/6}$$

where, R.N : Reynolds Number
V_c : Cruising Speed, Knots
C_r : Root Chord
ref : Reference Aircraft

The size ratio $\bar{L}(L_{bar})$ is $V_c * C_r /(1.63*10^4)$.

5. Approach Lift Coefficient ($C_{L,A}$) and Second Climb Lift Coefficient ($C_{L,2}$) : these are assumed to be 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, and 1.8.

The Input and Output Data can be summarized as follows;

Item No.	INPUT DATA	No.	OUTPUT DATA
1.	Passenger	1.	Lift / Drag Ratio
2.	Crews	2.	Altitude & sfc
3.	Range	3.	Payload
4.	Cruise Mach No.	4.	Gross Weight
5.	FAR TAKE-OFF & Landing field	5.	Empty Weight
6.	By-pass Ratio	6.	Fuel Weight
7.	Aspect Ratio	7.	Thrust
8.	Fineness Ratio	8.	Wing Area
9.	Fuselage Section vs Wing Area	9.	Fuselage Diameter
10.	Size Ratio	10.	Fuselage Length
11.	Lift Coefficient ($C_{L,A}$, $C_{L,2}$)	11.	Wing Span
12.	Others	12.	Others

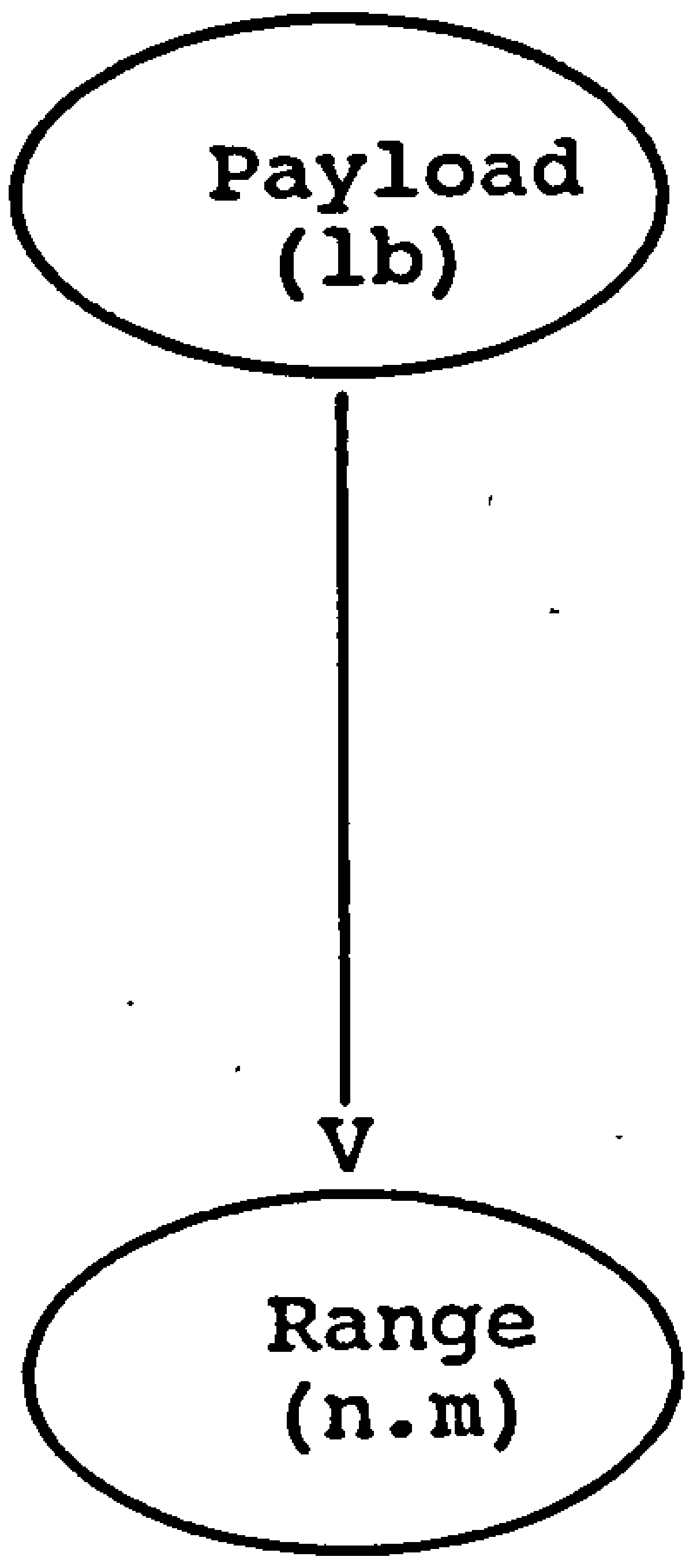
I.1.3 Design Process

1. Payload Range Process

From the data given in the specification, the payload range process is as follows;

PROCESS

EQUATION AND DATA



```
_CREW= _PAX / 30 + 2 pilots (*)

_cargo=_PAX * 50
_baggage=(_PAX+_CREW) * 30
_Payload=(_PAX+_CREW)*175
        +_cargo + _baggage

_climb_descent=100
_fuel_reserve=400
_design_range = _spec_range
              + _climb_descent
              + _fuel_reserve
```

2. Landing Performance

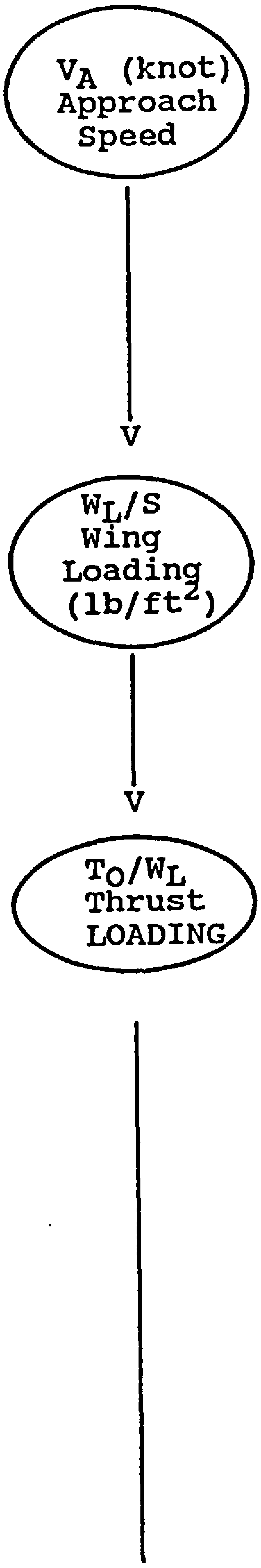
The specifications for most of the transport aircraft require a landing field length of 5000 ft on a standard day at sea level. The landing distance is measured horizontally from the point at 50 ft above the ground, with the approach speed not less than 1.3 times the stall speed, to the point where the aircraft stops completely on a hard, smooth, and dry runway. The FAR field length is the length obtained by dividing the landing distance by 0.6, as shown in the Figure I.1.3/1.

The missed approach should be considered in a landing manoeuvre and it occurs when the aircraft on the final approach to land cannot land for any of several reasons but climbs again with full power. Thus, the FAR for transport category aircraft requires sufficient thrust in case of the "Missed Approach" with one engine inoperative (OEI).

The Oswald Factor "e" explains a variation of Drag coefficient with lift coefficient for the entire airplane and it is termed the "Airplane Efficiency Factor". It will be considered in all the performance estimates.

(*) The items required for a calculation were converted in order for the variables to be used in the Turbo Prolog expressions either with the '_' prefixed or with the letter Capitalized as follows; _crew, _Crew, CREW, and so on.

PROCESS | **EQUATION AND DATA**



1. Landing Field Length

   Length $l_{T,L} = 0.3 * V_{A,2}$

2. Approach Lift Coefficient

   $C_{L,A} = 1.2, 1.3, 1.4, -- 1.8$

1. $V_A(\text{knot}) = 17.15*((W_{L/S})/(C_{L,A}))^{1/2}$

   $W_{L/S} = (V_A/17.15)^2*C_{L,A}$

   These wing loadings vary with the Approach Lift Coefficients.

1. Missed Approach Path-Angle Gradient (Radian / 100)

   : Following are F.A.R. Requirements.

Number of Engine	Gradient
2 not less than	2.1/100
3 not less than	2.4/100
4 not less than	2.7/100

2. Other Values

   Phi = 3.141592
   A.R. = Aspect Ratio
   Oswald Efficiency Factor e

Landing	Take-off	Cruise
0.7-0.75	0.7-0.75	0.85

3. Drag Coefficient is calculated as follows:

$$
\begin{aligned}
C_D = {} & \text{Zero Lift Drag } (C_{D,o}) \\
& + \text{Flap Effect } (C_{D,f}) \\
& + \text{Slat Effect } (C_{D,s}) \\
& + \text{Landing Gear } (C_{D,g}) \\
& + \text{Induced Drag} \\
& (C_L^2 / (Phi * A * e))
\end{aligned}
$$

C_D : Total Drag Coefficient

$C_{D,0}$: Zero Lift Drag Coefficient of aircraft in clean condition

$C_{D,f}$: An increment in profile drag coefficient due to trailing edge flap deflection. It is estimated to be 0.01, 0.02, and 0.03 for the flap deflection of 15, 25, 35 degree.

$C_{D,g}$: An increment in profile drag due to the landing gear extension coefficient due to the trailing edge.

- 0.015 during landing
- 0 during take-off

$C_{D,s}$: An increment in the profile drag coefficient due to the slat deflection. Assumed 0.

$C_{D,i}$: The Induced Drag Coefficient CL^2 / (PHI * A.R * e)

$$\frac{A_t}{S} = \frac{Phi*Dia^2}{S}\left[\frac{l}{d} - 1\right] + 3.38$$

S : Wing Area
Dia : Fuselage Diameter
A_t : Total Wetted Area
l : Fuselage Length
d : Fuselage Diameter

$$\frac{C_{D,o}}{C_{D,oref}} = \frac{A_t/S}{A_t/S_{ref}}$$

$C_{D,o}$: Aircraft to be
A_t/S Designed

$C_{D,oref}$: Reference Aircraft
A_t/S_{ref} to be selected

4. $T_O/W_L=(C_{D,o}/C_{L,A} + \text{Gradient})*(N/(N-1))$

N : Number of Engine
T_O : Total Engine Thrust
W_L : Landing Weight

In order for the climb gradient criteria to satisfy the condition with one engine operative, the required thrust to weight ratio with N number of engines all operating is expressed as above.

5. A Sample calculation is shown in Reference [7].

W_L/W_G

1. The ratio of Maximum Landing Weight to Maximum Gross weight at Take-off is required to be known at the landing situation equivalent to the take-off. The Following value are based on the existing aircraft.

Short Range	Mid Range	Long Range
0.73	0.82	0.91

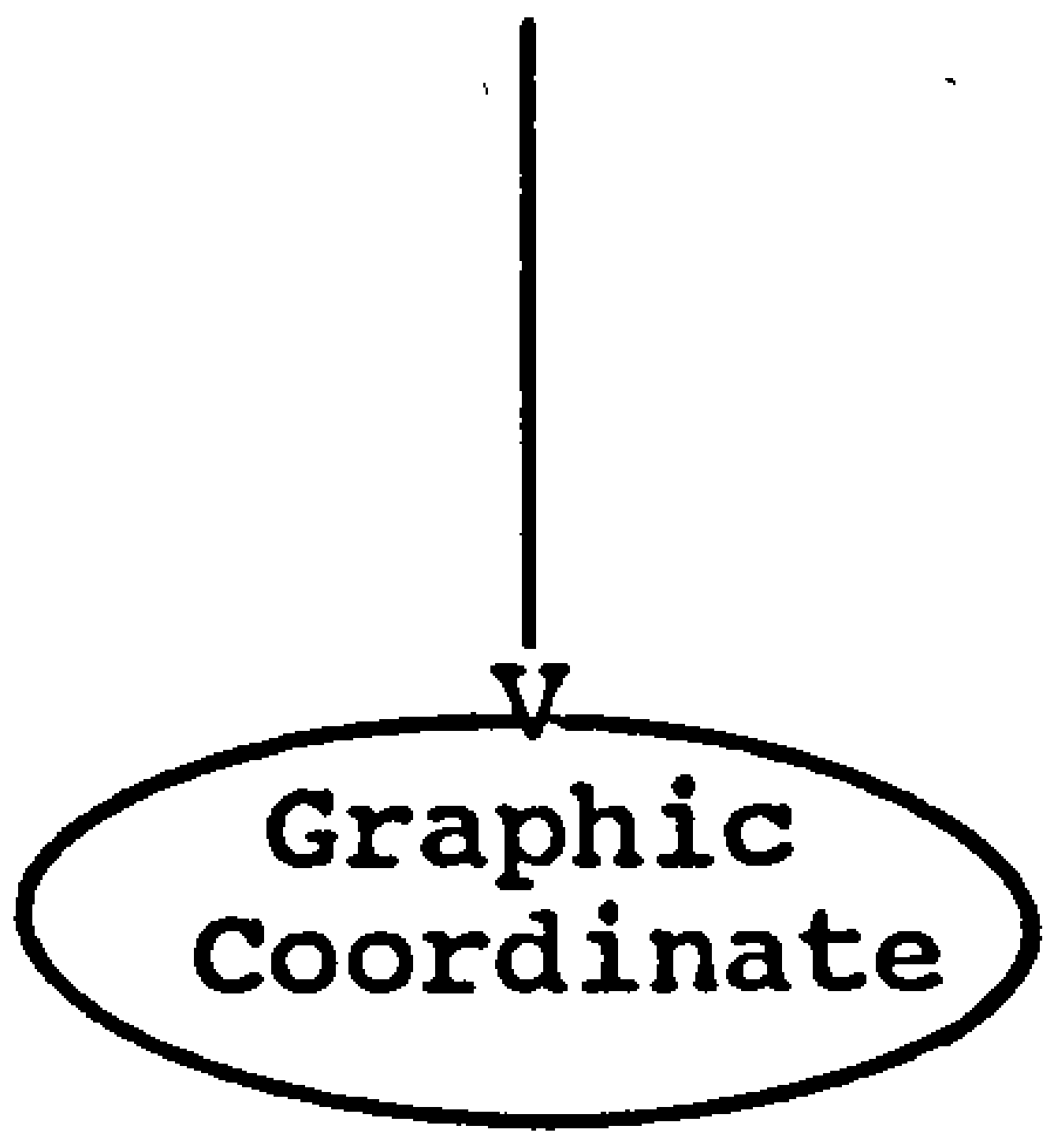
W_G/S

1. The Wing Loading is expressed in terms of weight ratio

$$\begin{aligned} W_G/S &= W_G/S * W_L/W_L \\ &= W_L/S * (W_G/W_L) \\ &= W_L/S \ / \ (W_L/W_G) \end{aligned}$$

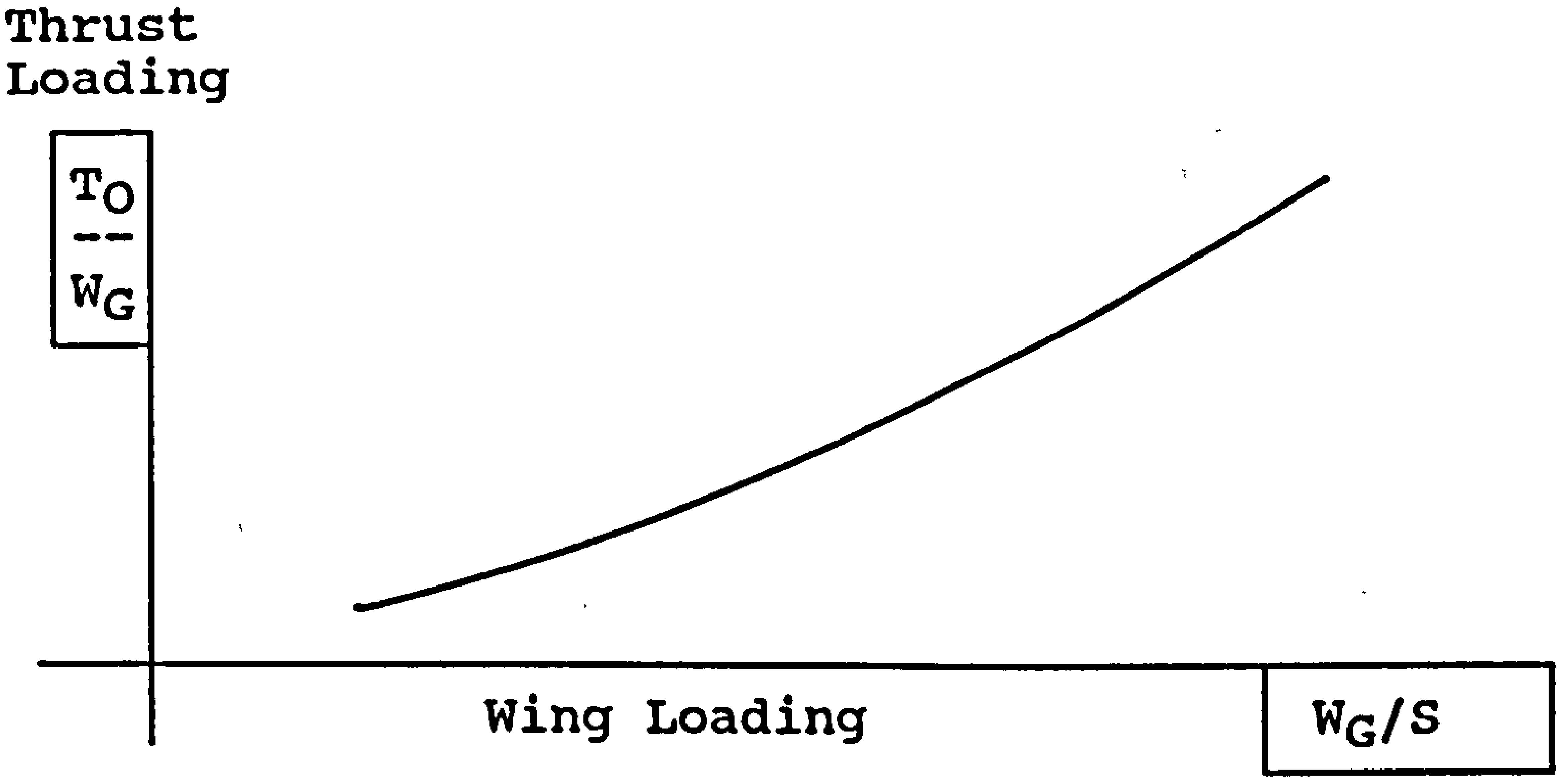
T_O/W_G

1. The Thrust Loading for landing equivalent to the take-off is expressed in terms of the weight ratio.



$$\begin{aligned} T_O/W_G &= T_O/W_G * W_L/W_L \\ &= T_O/W_L * W_L/W_G \end{aligned}$$

• $(x, y) = (W_G/S, T_O/W_G)$



3. Take-off Performance

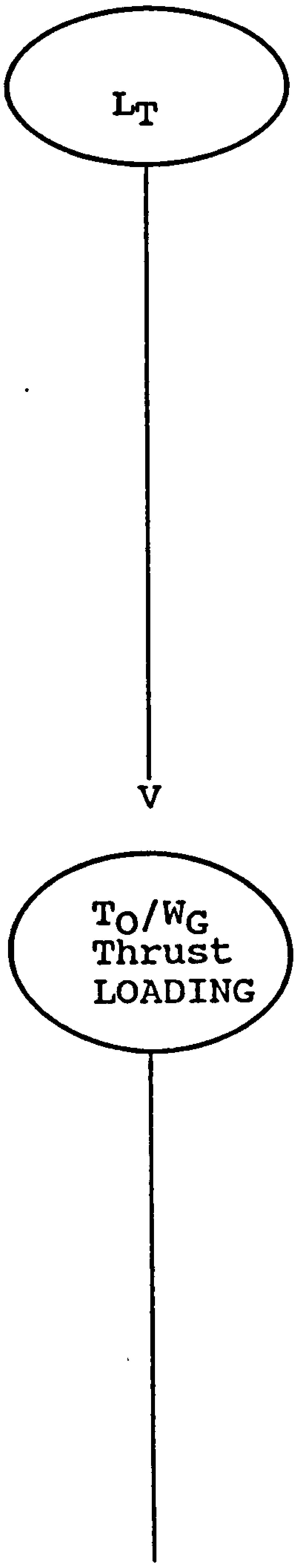
The Take-off Field Length in Figure I.1.3/2, often called "FAR balanced Take-Off Field Length" is defined as the distance from the point where the take-off run is initiated to the point where the aircraft reaches an altitude of 35 ft. The runway is assumed to be a smooth, hard, and dry surface. The length of runway is the distance where the aircraft reaches the decision speed V1, plus the distance from the point of decision speed V1 to the point where the aircraft stops completely. The decision speed V1 is the speed at which a pilot-in-command must decide whether he stops the aircraft or continues take-off. If the engine fails before the point of decision speed V1, aircraft stops on the runway; whilst the aircraft continues take-off if the engine fails at the speed greater than V1.

The value of Maximum Take-off coefficient $C_{L,T}$ was taken to be 1.44 times the lift coefficient of steady state second segment climb speed V2, since the speed V2 is defined as 1.2 times the stalling speed for the aircraft in take-off configuration. [7]

The second segment climb, where a flight following take-off is conducted at second segment climb speed V2 from an altitude 35 ft to 400 ft. It is FAR's requirement that sufficient thrust must be installed with one engine inoperative (OEI).

PROCESS

EQUATION AND DATA



1. The FAR requirement for Take-off field Length ranges over 5,000 to 11,000 ft.

$L_T = 37.6*(W_G/S)/(d.r. *C_{L,T}*(T_O/W_G)$

Density ratio, d.r.
At sea level,
$(1-0.02256*h(ft)/3280.84)^{4.2561}$

$C_{L,T} = 1.44 * C_{L,2}$

$C_{L,2}$ (given) = 1.2, 1.3,--, 1.8

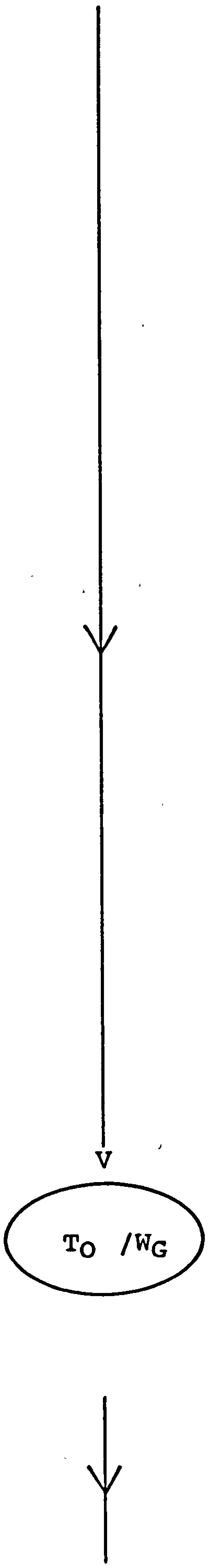
1. The Second Climb Segment Gradient (Radian / 100)

: Following are F.A.R. Requirements.

No. of Engine		Gradient
2	not less than	2.4/100
3	not less than	2.7/100
4	not less than	3.0/100

2. Other Values

Phi = 3.141592



A.R. = Aspect Ratio

Oswald Efficiency Factor e

Landing	Take-off	Cruise
0.7-0.75	0.7-0.75	0.85

3. The Drag Coefficient can calculated as follows;

C_D = Zero Lift Drag ($C_{D,o}$)
+ Flap Effect ($C_{D,f}$)
+ Slat Effect ($C_{D,s}$)
+ Landing Gear ($C_{D,g}$)
+ Induced Drag
($C_{L,T}^2$ / (Phi* A * e))

$$\frac{A_t}{S} = \frac{Phi*Dia^2}{S}\left[\frac{1}{d} - 1\right]+3.38$$

S : Wing Area
Dia : Fuselage Diameter

$$\frac{C_{D,O}}{C_{D,Oref}} = \frac{A_t/S}{A_t/S_{ref}}$$

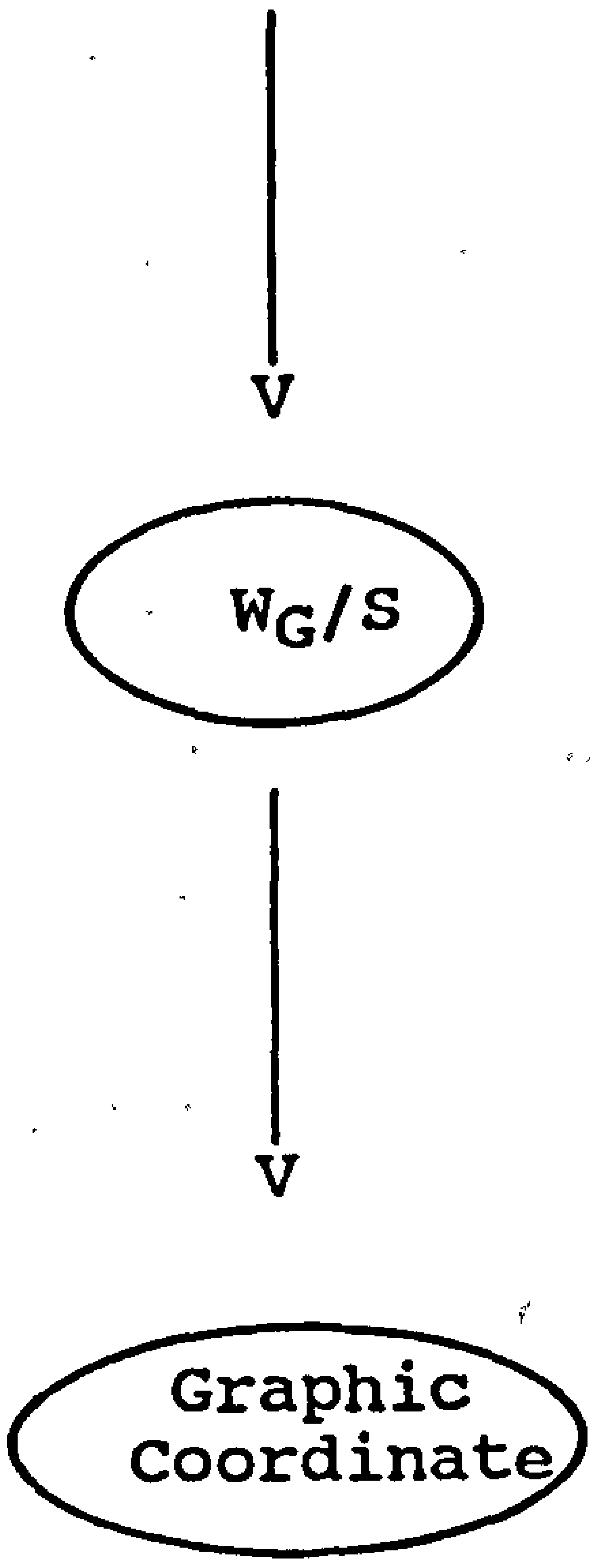
$C_{D,O}$ A_t/S : Aircraft to be Designed

$C_{D,Oref}$ A_t/S_{ref} : Reference Aircraft selected

4. T_O/W_G= ($C_{D,O}$/ $C_{L,2}$ +Gradient) * (N/(N-1))

N : Number of Engine

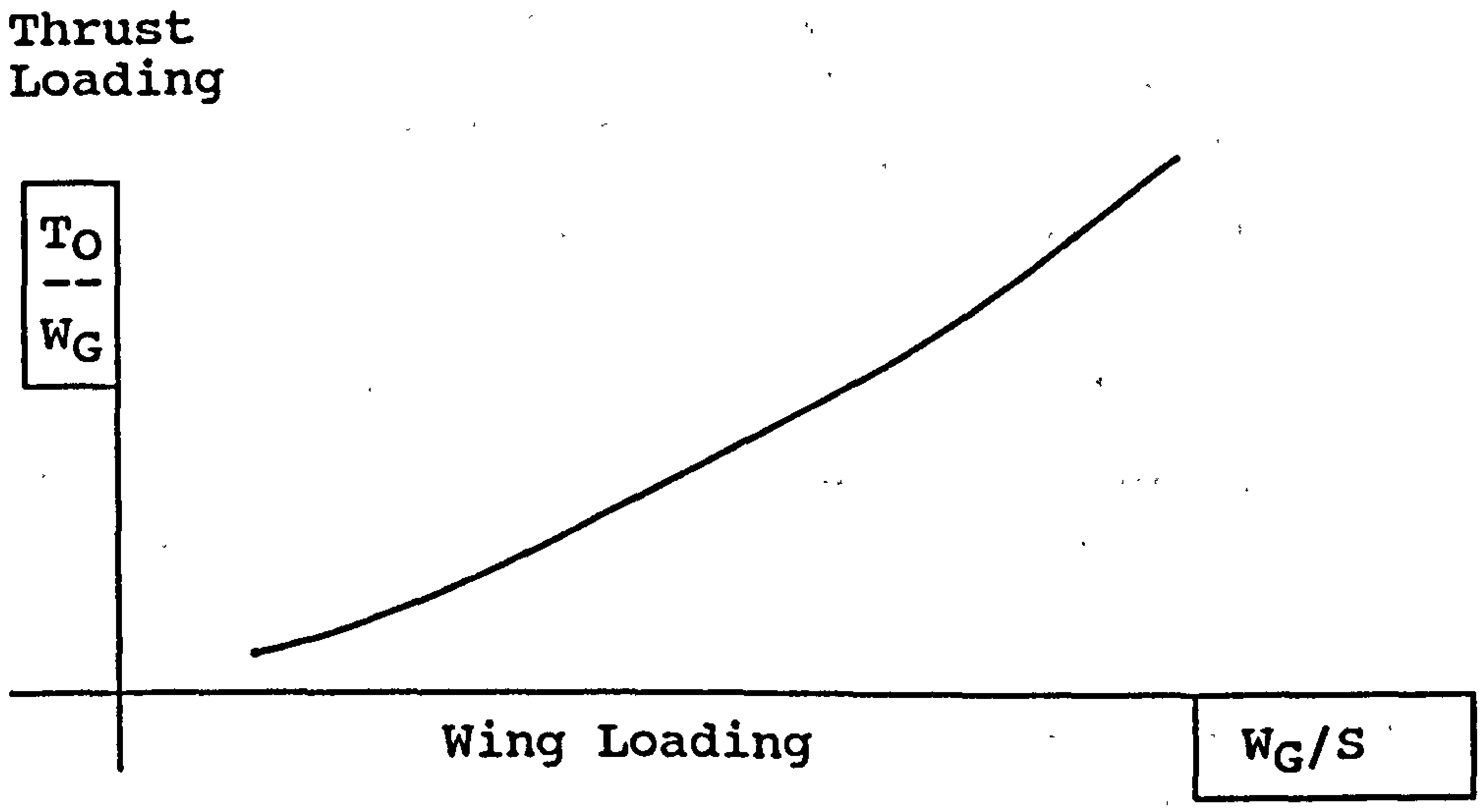
In order for the climb gradient criteria to satisfy the condition- with one engine operative, the required thrust to weight ratio with the N number of engines all operating is expressed as above.



5. A Sample calculation is shown in Reference [7].

1. $W_G/S = L_T/37.6 * d * C_{L,T} *(T_O/W_G)$

- $(x, y) = (W_G/S, T_O/W_G)$



4. Cruise Performance

The Cruise Performance analysis matches the performance characteristics between the relevant engine and the airframe to achieve the specified design range at a given cruising speed (Knot or Mach No.) with the fuel consumption minimized. The Well-known Breguet range equation represents the quantitative relationship incorporating engine, airframe, and fuel during cruising flight.

$$\text{Range}=[V_{cr} * (L/D) / c] * \text{Log}_e [1/ \{ 1- (W_f/W_G)\}]$$

$$B \text{ (Breguet Factor)}=V_{cr}* (L/D) /c = a * M * L/D / c$$

R : Cruise Range nautical mile.
V_{cr} : Cruise Speed(Knot, 1 kt=1 nm/h=1.1508 mph=1.852 km/h)
L/D : Aircraft Lift to Drag Ratio
c : Engine Specific Fuel Consumption, lb/lb/hr (Pounds of Fuel per Pounds of Thrust per Hour)
W_G : Aircraft Gross Weight
W_f : Aircraft Fuel Weight
a : Speed of Sound, knot
M : Cruising Mach Number

In the above equation, the fuel consumption (or fuel fraction, W_f / W_G) for specified range becomes a minimum as the Breguet Factor becomes a maximum. The Breguet Factor for a specified cruising speed becomes a maximum when the Lift-to-Drag Ratio becomes a maximum. Thus, it is important to get as high a Lift-to-Drag ratio as possible. The Lift-to-Drag Ratio can be expressed as follows;

$$L/D = C_L / \{ C_{D,0} + (C_L^2 / (\text{Phi} * \text{A.R.}* e))\}$$

The Maximum L/D ratio occurs when $C_{D,0}$ equals $C_L^2 / (\text{Phi} * \text{A.R.}* e)$. That is, $L/D_{max} = (1/2) * (C_L / C_{D,0}) = (C_{D,0} * \text{Phi} * e)^{0.5} /(2 * C_{D,0}) = (1/2) * ((\text{Phi}*A*e)/C_{D,0})^{0.5}$.

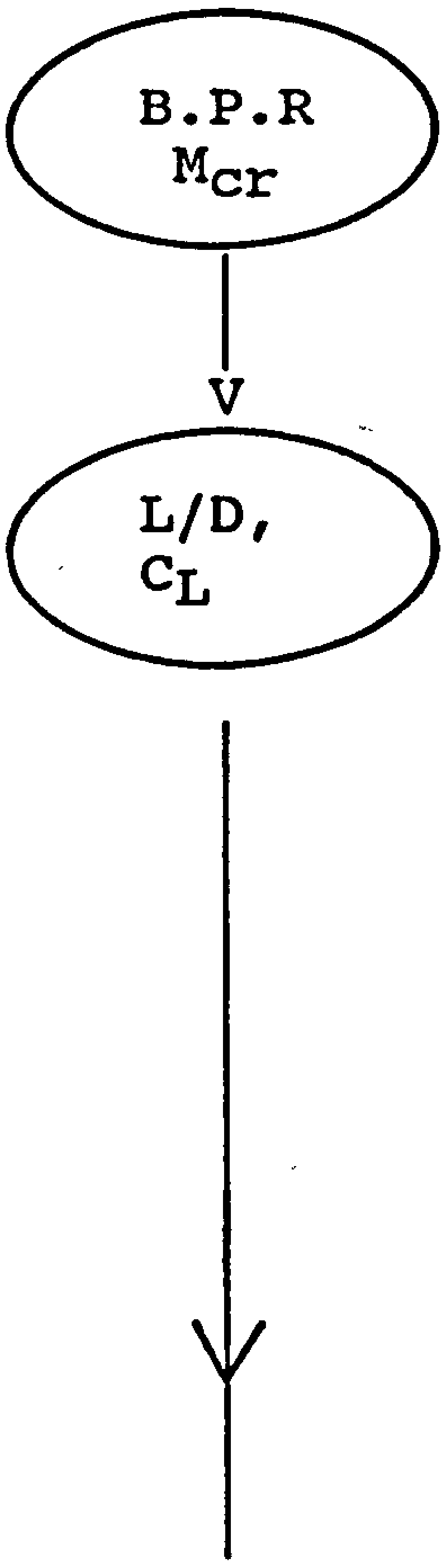
The Oswald efficiency factor e is assumed to be 0.85 for a clean configuration of Jet Powered aircraft. It is often needed to get other Lift to Drag ratio. So, an alternate L/D used here is 0.97 times L/D_{max}.

Another consideration is the engine's capability to provide a sufficient thrust for balancing the aircraft drag in the cruising flight at the desired altitude and Mach No. of aircraft operating condition. The thrust available varies depending upon altitude, Mach No., and Engines. To get a thrust loading (T_O/W_G) in the equation $T_O/W_G = 1/ \{ (T_C/T_O) * (L/D_{max})\}$, the maximum lift to drag and the ratio of maximum thrust (T_C) to take-off thrust (T_O) must be known first.(*)

The By-pass ratio is the engine parameter chosen by a designer. However, the altitude can be obtained by considering the cruise speed, lift coefficient, and various wing loadings. Thus a pair of wing loading and its associated thrust loading can be obtained according to the 2 conditions such as one maximum lift to drag ratio and the other lift drag ratio. However, an off-design cruise condition shall not be investigated further.

PROCESS

EQUATION AND DATA



1. Described in Specification

1. Ratio of Lift to Drag

$$L/D_{max} = 0.5*(Phi*A*e/C_{D,0})^{0.5}$$

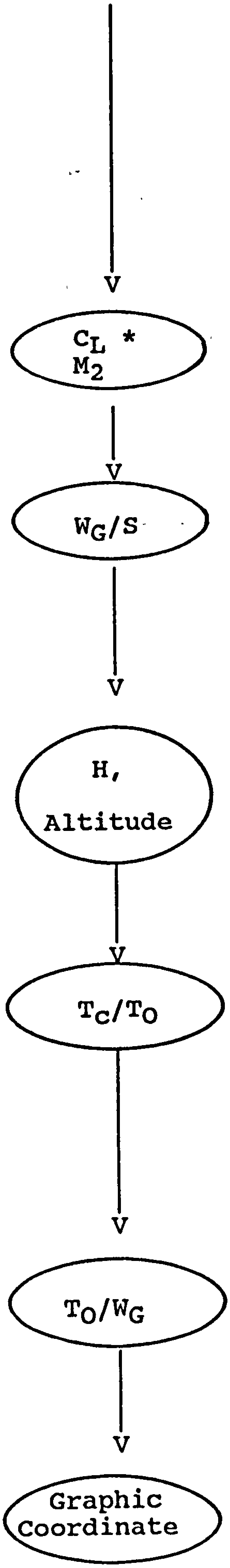
Phi = 3.141592
A = Aspect Ratio

e : Oswald Efficiency Factor e

Landing	Take-off	Cruise
0.7-0.75	0.7-0.75	0.85

$$C_{L,M} = (Phi * A.R* e * C_{D,0})^{0.5}$$

(*) The graph for T_C/T_O can be found in Page 133-136 of the reference [7].



Here, let's assume the L/D ratio compared with Maximum Condition.

$L/D_{assume} / L/D_{max} = k$

$K=2/k$

$C_{L,K}=C_{L,M}*((K-(K^2-4)^{0.5})/2)$

- Can be decided from the above

- Assume the wing loading variation

  80, 90, 100, 110, 120, 130,--160

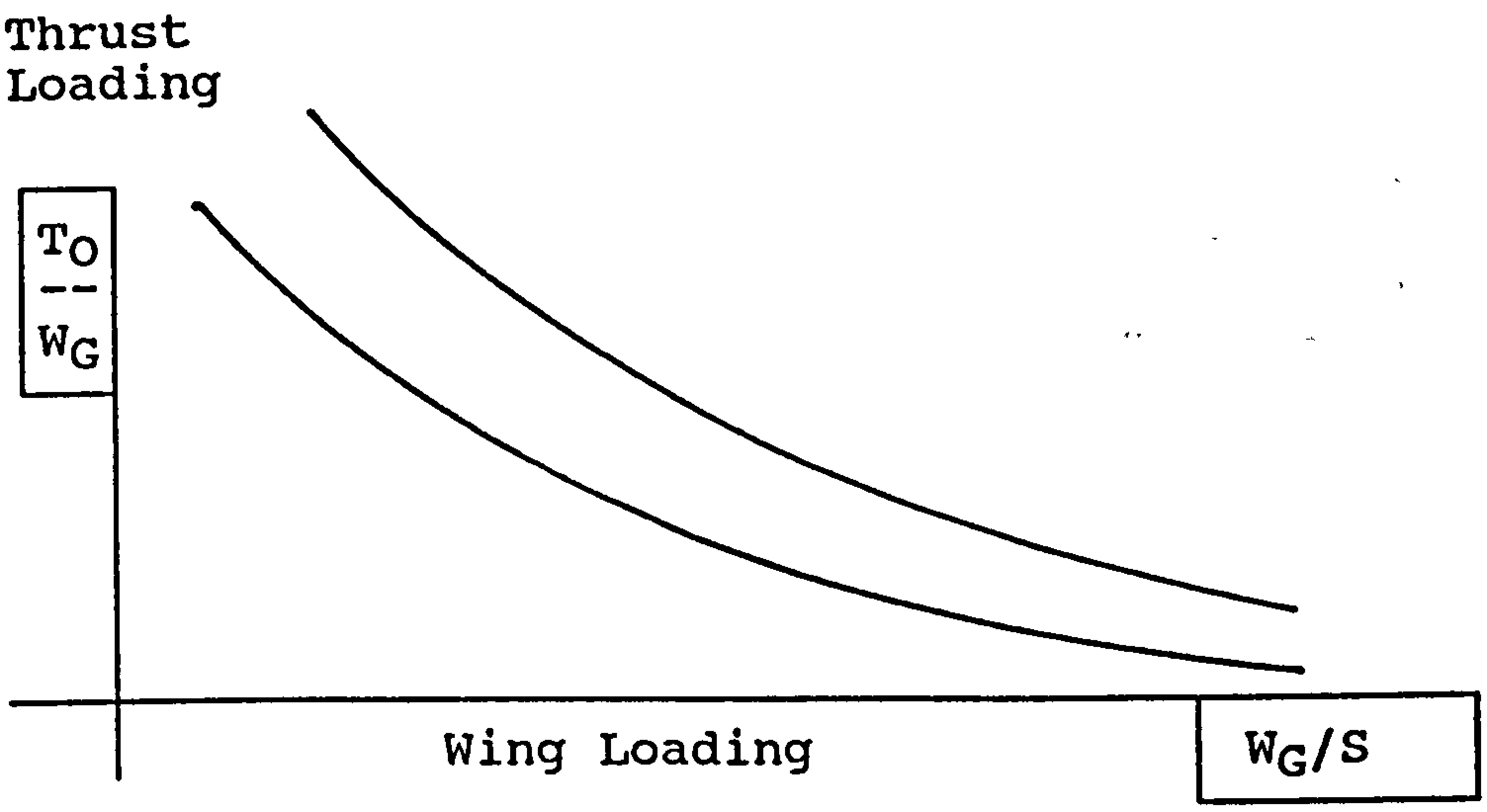
- $H = (1-(W_G/S/(C_L*M^2)/1481.34))^{1/5.2561} * 145,427.3$

- This can be achieved from the graph [7] by using the relation among the Altitude, Mach No., and S.F.C .

- $T_O/W_G = 1/ (L/D * T_C/T_O)$

- $(x, y) = (W_G/S, T_O/W_G)$
- This is drawn for the 2 cases, L/D_{max} and L/D assumed.

$$L/D_{\text{assumed}} = L/D_{\max} * 0.97$$



5. Aircraft Matching and Sizing Procedure

The Aircraft Matching Procedure is a step for a designer to plot all coordinates of wing loading and thrust loading for a landing performance, take-off performance, and cruise performance and to find the intersection points among them. From these intersection points, a sizing procedure starts to find a maximum gross weight (W_G), payload (W_P), fuel weight (W_f), empty weight (W_e), fuselage diameter, wing area, wing span, etc.

Let's assume, $W_G = W_P + W_f + W_e$,

$$1 - (W_e / W_G) = (W_P + W_f) / W_G = \overline{U},$$

$$W_G = W_P / (U - (W_f/W_G)).$$

W_G : Gross Weight, lb
W_f : Fuel Weight, lb
W_P : Payload Weight, lb
W_f/W_G : This fuel fraction can be obtained from a cruising performance.

U bar : The Useful load fraction is presented in the Figure I.1.3/3. This trend is the result of investigating about 40 existing aircraft varying the gross weight ranging 10,000 lb to 800,000 lb and the thrust to weight ratio ranging from about 0.23 to 0.46.

$$\bar{U} = - 1.402 \; (T_O/W_G) + 0.769$$

Then we can get a Gross Weight (W_G). A Fuel weight can be solved from the Breguet Equation, $W_f/W_G = 1- (1/e^{R/B}))$. In Breguet factor $B = V_{cr} * (L/D) \; / \; c$, the Lift to Drag ratio must be estimated precisely. The following procedure explains this very well.

$$\begin{aligned} C_{L,final} &= C_{L,final} * C_{L,initial} / C_{L,initial} \\ &= C_{L,initial} * C_{L,final} / C_{L,initial} \\ &= C_{L,initial} * (W_L / W_G) \; ....(\text{Eq. } 1) \end{aligned}$$

initial : at initial cruise
final : at final cruise

As the lift coefficients at an Initial cruise and at a Final Cruise are different, the Lift to Drag ratio changes accordingly. Therefore, it is reasonable to use an average Lift to Drag ratio, $L/D_{average}$.

$$\begin{aligned} L/D_{average} &= (1/2) * \{ L/D_{initial} + L/D_{final} \} \\ &= (1/2) * L/Dmax * \left\{ \frac{L/D_{initial}}{L/D_{max}} + \frac{L/D_{final}}{L/D_{max}} \right\} \\ &= (1/2) * L/Dmax * (K0 + K_{final}) \end{aligned}$$

$$K0 = \frac{L/D_{initial}}{L/D_{max}} , \quad K_{final} = \frac{L/D_{final}}{L/D_{max}}$$

From equation (Eq. 1),

$$\frac{C_{L,\, final}}{C_{L,\, max}} = \frac{C_{L,initial}}{C_{L,\, max}} * (W_L / W_G)$$

Then, $\bar{C}_{L,\, final} = \bar{C}_{L,\, initial} * (W_L / W_G)$.

As weight ratio W_L/W_G can be obtained in the Figure I.1.3/4, $C_{L,\, initial} / C_{L,\, max}$ can be achieved according to K0 easily from the Figure I.1.3/5.

From the intersection points of Wing loading, thrust loading, and the known Gross weight, the designer can get an Wing Area(S), Take-off Thrust(T_O), etc. Also with the ratio of a fuselage cross section to a wing area known from the procedure of estimating the zero lift drag $C_{D,0}$ with respect to the reference aircraft, a fuselage diameter can be obtained from the following relationship.

$$A_f/S = (Phi * Dia_2) * (1/S)$$

A_f : Fuselage Cross Section
S : Wing Area
Dia : Fuselage Diameter

From a relationship among the aspect ratio, wing area, and fineness ratio, the dimension of an wing span and fuselage can be found as follows;

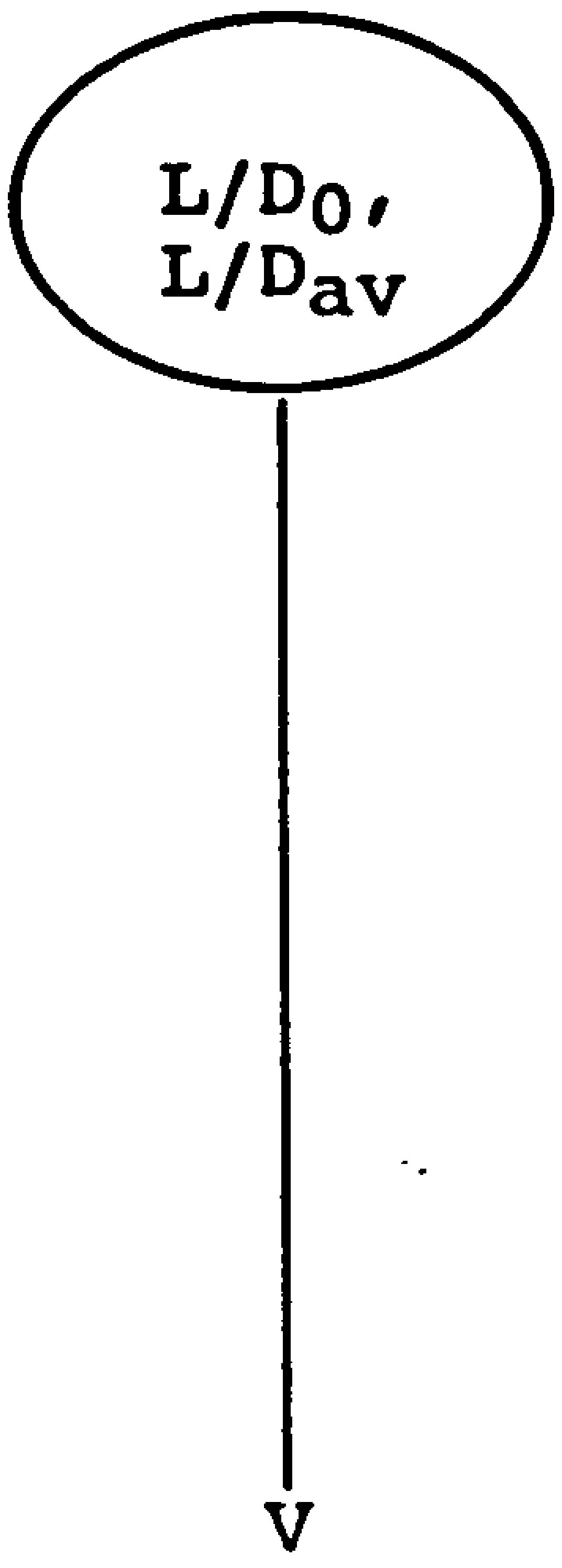
$$b^2 / S = A.R., b= (S * A.R.)^{0.5}$$

$$l/d = Fineness\ Ratio$$

As the above estimated parameters are closely related with the parameters in other design areas, interactions among processes are inevitable and therefore trade-offs follow when conflicts are encountered.

PROCESS

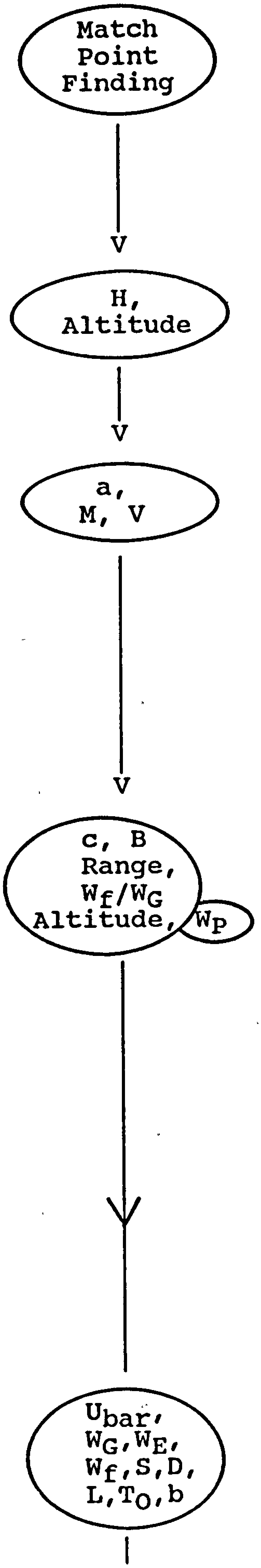
EQUATION AND DATA



- To get a more exact value for cruising condition, the L/D ratios at the beginning and the end of cruising flight are considered and their average is used.

The calculation procedure is described in reference [7].

$$L/D_{av} = (L/D_0 + L/D_{final}) / 2$$



- The intersection points among the graphs of Landing, Take-off and Cruising conditions.

- $H = (1-(W_G/S/(C_L*M^2) /1481.34))^{1/5.2561} * 145,427.3$

- Speed of Sound
  h>=35,600 a=576.5 knot
  h< 35,600 $a=1.9438 * (115,798.76 - 0.795564*H(ft))^{1/2}$

- Velocity = a * M

- Specific Fuel Consumption c,

  c = function of Mach No., By-pass Ratio [7]

- Breguet Factor

  $B = (L/D_{av} * V) / c$

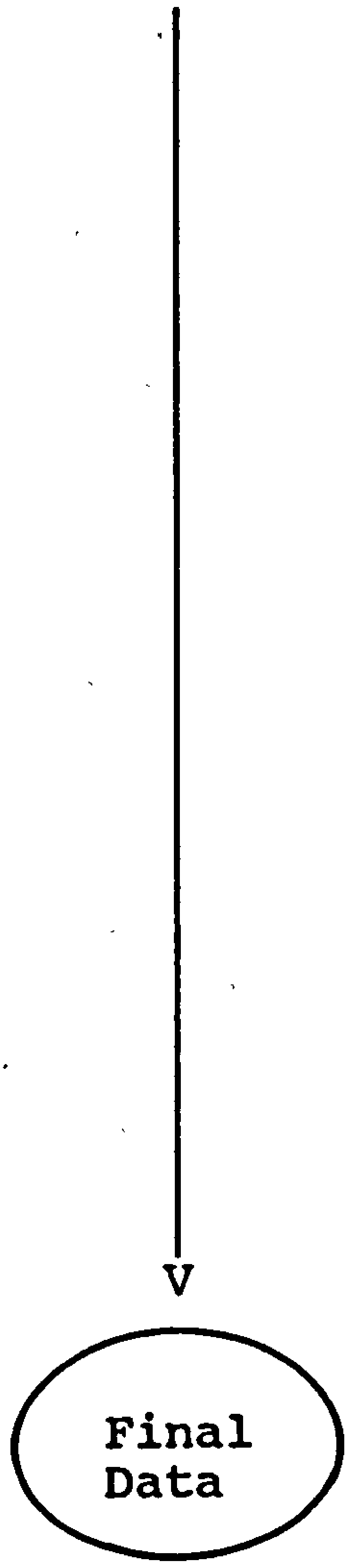
  $W_f/W_G = 1 - (1/(e^{R/B}))$

- W_P, From Payload_Range Process

- Useful Load Fraction, U_{bar}

  $U_{bar} = -1.402(T_O/W_G) + 0.769$

- Gross Weight,



$W_G = W_P / (U_{bar} - W_f/W_G)$

- Empty Weight,

  $W_e = W_G * (1 - W_f/W_G) - W_P$

- Fuel Weight

  $W_f = (W_f/W_G) * W_G$

- Wing Area and Take off Thrust,

  $S = W_G / (W_G/S)$
  $T_O = (T_O/W_G) * W_G$

- Fuselage Diameter

  $D = (A_f/S * 4/Phi)^{1/2}$

- Length = Fineness Ratio * D

- Span = $(S * A.R)^{1/2}$

- Gross Weight, Empty Weight, Fuel Weight, Wing Area, Take-off Thrust, Fuselage Diameter, Fuselage Length, Wing Loading, Thrust Loading.

I.2 Wing Design Analysis

I.2.1 Detailed Analysis Procedure

1. Two Dimensional Wing Design

This determines an airfoil section shape giving due consideration to the cruise Mach No. and cruise lift coefficients. Supercritical airfoils have been investigated for high subsonic transport purposes and the RAE series of airfoils were analyzed for appropriate selection. [1]

1. The 3 RAE series of supercritical airfoils are

- RAE 9515 : 10.5 % Thickness Ratio
- RAE 9530 : 10.5 % Thickness Ratio
- RAE 9550 : 12.2 % Thickness Ratio

2. As independent factors, Cruise Mach No. (specified in specification) and cruise Lift / Drag Ratio (estimated in parametric study) have influences on the dependent factors such as Lift Coefficient, Stall Behaviour, Cruise Lift/Drag ratio, Pitching Moment Coefficient, and 2 Dimensional Drag rise.

The **Lift Coefficients** at Low speeds and a high incidence should be high enough to avoid an early stall and this characterises the maximum Lift Coefficient at a low Mach Number. The higher the lift coefficient, the better the low speed characteristics.

The **Stall behaviour** is a phenomenon where the lift coefficient drops suddenly just after the stall and it is measured as the slope over 1 degree of incidence after the stall at low speeds. The higher the stall lift coefficient, the better stall behaviour.

The **Cruise Lift to Drag ratio** should be as high as possible and therefore minimum drag coefficient results. The Pitching moment should be low and the two dimensional drag rise refers to an increase in drag due to the compressibility effects. The critical Mach No. should be sufficiently high to avoid theses effects.

2 Three Dimensional Effects

To achieve an acceptable design in terms of the three dimensional drag rise, aeroelastics, tip stall, flap effectiveness, and weight, a range of swept angels from 15 degrees to 45 degrees, increasing by 5 degree, are evaluated.

The appropriate thickness ratio is also determined for suitable drag rise characteristics and unsuitable swept angles are eliminated.

1. The Three Dimensional drag rise Mach No. is caused by the appearance of the fuselage which reduces the Drag rise Mach Number by 0.02 to 0.05 and must be accounted for. The designer can select the decreased amount from a similar type of aircraft to the one being designed.

AIRCRAFT	Decreased Magnitude(D_{ecr})
BAE 125	0.05
Boeing 747	0.035
Boeing 727	0.02

Thus the relations are as follows;

M_D /\=0 = Mach No.economic cruise + D_{ecr}

M_D = M_D /\=0 / cos $^{1/2}$ /\$_{1/4}$

As an example, the wing design analysis in chapter 4 shows how to select the appropriate airfoil, that is, the best one and the second best one.

2. The high speed requirement must be checked by

   1. Max M $_{cr}$ - Mach No. $_{3\text{-}D}$ drag rise =< 0.02
   2. Eliminate the thickness ratio over 18 %.

3. The aeroelastic stiffnesses such as torsion and bending of a wing are checked and are different depending upon the position of engines (wing mounted engine or rear fuselage engine) and the incorporation of active control technologies.

   1. Torsional Stiffness

$$\frac{A.R.^{3/2}}{(t/c)^2} < \frac{N * 10^8}{V_D * \cos \Lambda_{1/4}}$$

Load Factor N : 3 for wing mounted engine
: 2.5 for non wing mounted engine

V_D : Design Diving Speed

= (M_{max} + 0.05) * Speed of Sound

   2. Bending Stiffness

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

N : Ultimate Load Factor

3.75 for Large Transport
2.5 for Active Control

4. The tip stall is checked by evaluating the spanwise airload distribution using the combination of the Schrenk and Stanton - Jones formula. However, for simplicity in application, the effects from the wing twist, taper ratio, and camber effects are not considered.

- Schrenk's approximate Method:Basic Load Distribution

$$\frac{C_L(y)/\overline{C_L}}{c(y)/\bar{c}} = \frac{K * a_0 * (\alpha_0 + \epsilon) / \overline{CL}}{3(n*\tau^2-n+\tau+1)/\ 2*(\tau^2+\tau+1)}$$

- Additional Load Distribution : Stanton - Jones Formula

For $n < 0.7$

$$\frac{C_L(y)/\overline{C_L}}{c(y)/\bar{c}} = 1.28(1-n^2)^{.5} + (14.13\ n - 6.35)(\bar{y}-0.425)$$

For $n >= 0.7$

$$\frac{C_L(y)/\overline{C_L}}{c(y)/\bar{c}} =1.28(1-n^2)^{.5}+(4.25-53.8(n-.815))(\bar{y}-0.425)$$

where

$\bar{y} = 0.42+A*m((4.4+5\tau)*\tan \Lambda_{1/4} /m +10.4\tau^{.5}-6.7)/10^3$

C_L : to be input by the designer

m = $1 - M^2$

M : Mach No. economic cruise

α_0 : -0.18 degrees(taken from RAE Airfoil section)

K : Correction Factor (0.5)

ϵ : $- n * \epsilon_0$

τ : Taper Ratio to be input by designer

- Local lift coefficient along the span must be greater than Cruise Stall Lift Coefficient.

5. The Flap effectiveness is reduced as a wing increases its swept angle and the Maximum Lift Coefficient defined by a designer should be greater than the Landing lift coefficient multiplied by COS $/\backslash_{1/4}$. The Lift Coefficient is increased up to 1.65 times a Lift coefficient due to the leading edge device and accordingly becomes near 3.

- C_L max =< C_L landing * COS $/\backslash_{1/4}$
- C_L increases upto 1.65 * C_L. That is due to both 1.15 * C_L for a basic wing and 0.5 for leading edge devices.

6. The Wing weight is measured by considering effects from the swept angle, taper ratio, range, thickness ratio, wing area, aspect ratio, etc. When composite material is to be incorporated, the weight can be reduced down to the 75 % of the metal wing weight.

$$W_{wing} = C1 * \left[\frac{b*S}{\text{Cos } /\backslash} \; \frac{(1+2*T.R)}{(3+3*T.R)} \left[\frac{M*N}{S} \right]^{0.3} \left[\frac{V_D}{Tau} \right]^{0.5} \right]^{0.9} \text{kgs}$$

b : Wing Span (m)
S : Wing Area (sq. m)
/\ : Swept angle at quarter chord
T.R : Taper Ratio
M : Aircraft Gross Weight (kg)

N : Design Load Factor
V_D : Design Diving speed (m/sec EAS)
Tau : Thickness Chord Ratio at the root.
C1 : 0.028 for long range
: 0.034 for short range

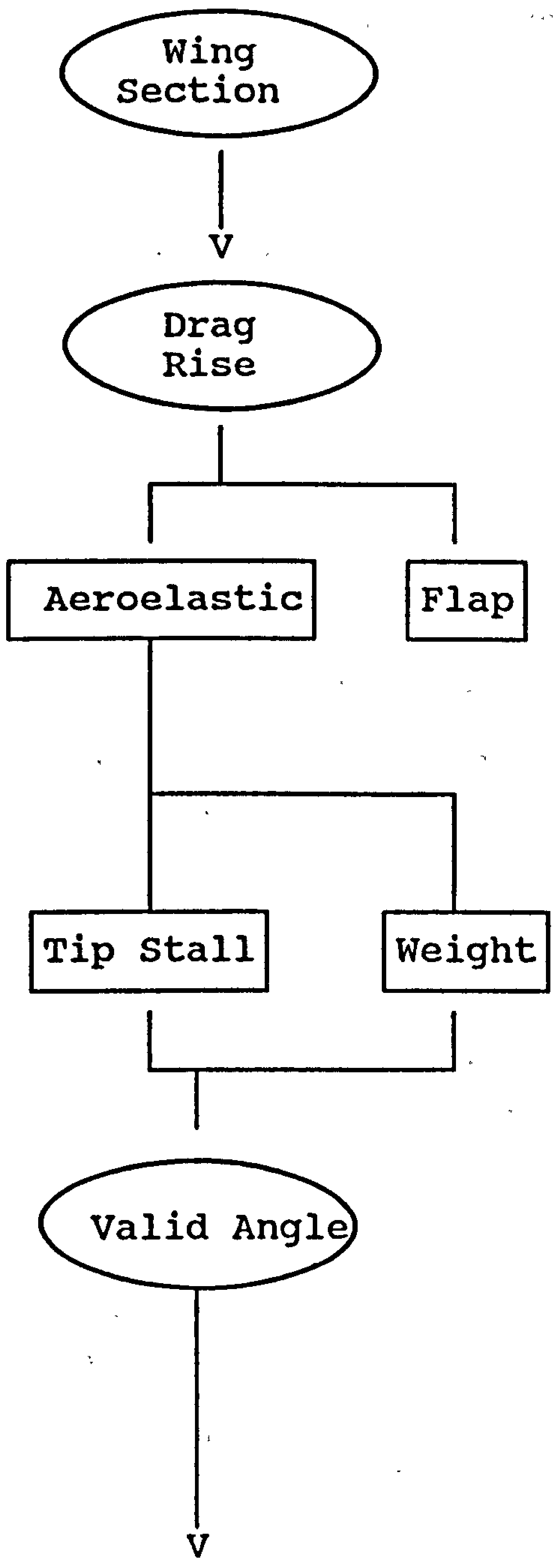
I.2.2 Input and Output Data

Input Data	Output Data
1. From specification . Cruising Mach No. . Range . Aspect Ratio 2. From Parametric Study . Wing Loading . Thrust Loading . Weight(W_E, W_P, W_G) . Swing . Span, b . L/D ratio . CL cruise 3. Values to be input . Taper ratio (0.2 to 0.6) . Twist Angle . Dihedral Angle . Fuselage Wing interaction 4. Case to be considered . Engine Position . Active Control . Composite Use	1. Supercritical Airfoil 2. Thickness Ratio 3. Chord Length C_r =(2/(1+T.R.)*S / b C_t =T.R.* C_r t_r=C_r * $(t/c)_r$ t_t=C_t * $(t/c)_t$ 4. Swept Angle 5. Twist, Dihedral Angle 6. Weight . per engine position . per active control . composite or not

I.2.3 Procedures

PROCESS

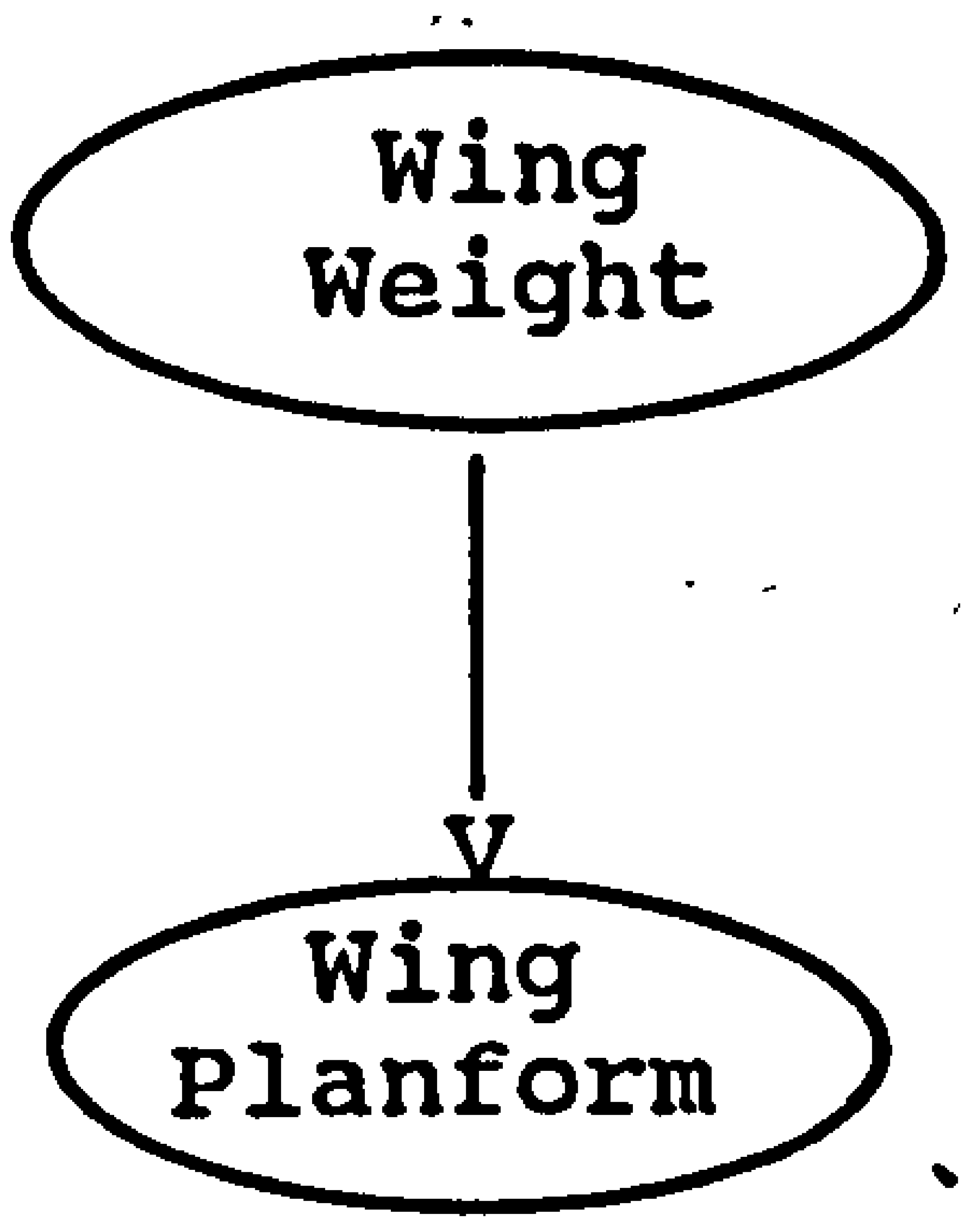
EQUATION AND DATA



1. RAE 9515,9530, 9550
   (Selecting examples refer to the Figure I.2.3/1)

1. Fuselage Wing interaction

(described as above)



Aspect Ratio, Wing Span, Wing Area, Swept Angle, Twist, Taper Ratio, Dihedral Angle.

I.3 Fuselage Design Analysis

I.3.1 Design considerations

The fuselage is the component which must accommodate the passenger comfortably and hence fuselage design is generally started from the inside outward. Thus the dimensioning of Civil Transport fuselage should be given the following considerations.

1. The number of seats abreast which is selected from 4, 5, 6, 7, 8, and 9 seats according to the Number of Passengers. The next is to determine seat dimensions and the Number of aisles.

   The fuselage cabin has one or two aisles depending upon the Numbers of Passengers. For example, one aisle is selected for passenger numbers below 179. Over 179 passengers, 2 aisles are used. [3],[4] Also 2 decks are feasible for over 500 passengers. This was also classified in detail in the specification study in section I.1.1.

   Fuselage thickness is assumed to be either about 4 inches or a dimension which is 0.02 * Fuselage Width (or internal diameter) + 1 inch (or 2.54 cm).

2. Seat pitch is closely related to passenger comfort and it depends upon the designer's choice. The entrance and exit doors, including the emergency doors must be considered. The Galley / Lavatory / Wardrobe can be arranged accordingly.

The nose fuselage, including flight deck is of a typical length, 1.5 to 2 times the fuselage diameter.

The rear fuselage (or tailcone), where the horizontal or vertical stabilizer will be installed, is of a typical length, 2.5 to 3 times fuselage diameter and a half of the tailcone angle is 10 to 12 degrees. These factors determine the overall length and this length associated with diameter determines fineness ratio.

I.3.2 Input and Output Data

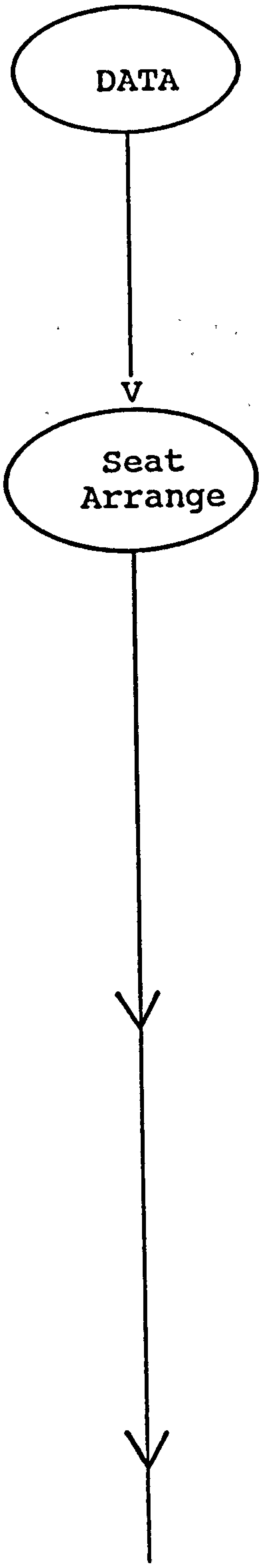
Input Data	Output Data
1. From specification . No. of Passengers . Range 2. From parametric study . Length / Diameter (Fineness Ratio) . Phi * Dia^2 / 4 * S (Cross section vs Wing Area) . V_D, Design Diving speed 3. Others . Seat, Pitch/Width . Aisle Width . Seat Dimension . Fuselage thickness . Others	1. Revised Length 2. Revised Diameter 3. Phi * Dia^2 / 4 * S (*) 4. Fuselage Weight 5. Seat Dimension & Seat pitch 6. Aisle Width 7. Nose Fuselage Length 8. Tail Fuselage Length

---

(*) This is the ratio of fuselage cross section to wing area.

I.3.3 Detail Design Procedure

PROCESS | **EQUATION AND DATA**



1. Data

- Fineness Ratio
- Length
- Diameter
- No of passengers
- Range

1. Number of seat abreast

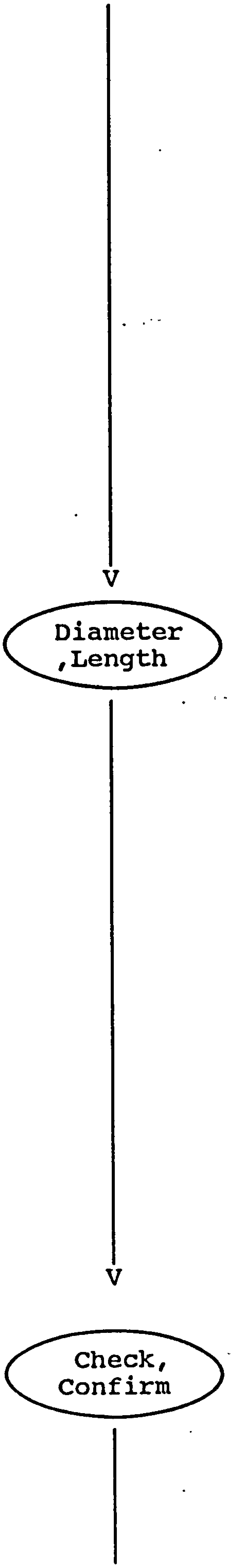
Passenger	Seat Abreast
up to 65	4
66 - 130	5
131 - 260	6
261 - 420	8
over 421	9

2. Number of Aisles

Passenger	Aisles
up to 65	1
66 - 179	1
over 180	1

3. Number of Access Doors

Passenger	Access Door
up to 80	1
66 - 179	2
over 180	3



4. Fuselage Thickness

   4 to 6 inches or
   $0.02 * Dia_{int} + 1$ inch

5. Gap between seat nearest window and fuselage : about 2 inch

6. Seat pitch & dimensions (Normal/Economy Class)

   . seat width 16.5 to 17"
   . seat pitch 28 to 31"
   . armrest width 2 to 2.25"
   . Aisle width minimum 15"

   * Seat pitch is added 10" where access door is positioned.

1. Fuselage Diameter,

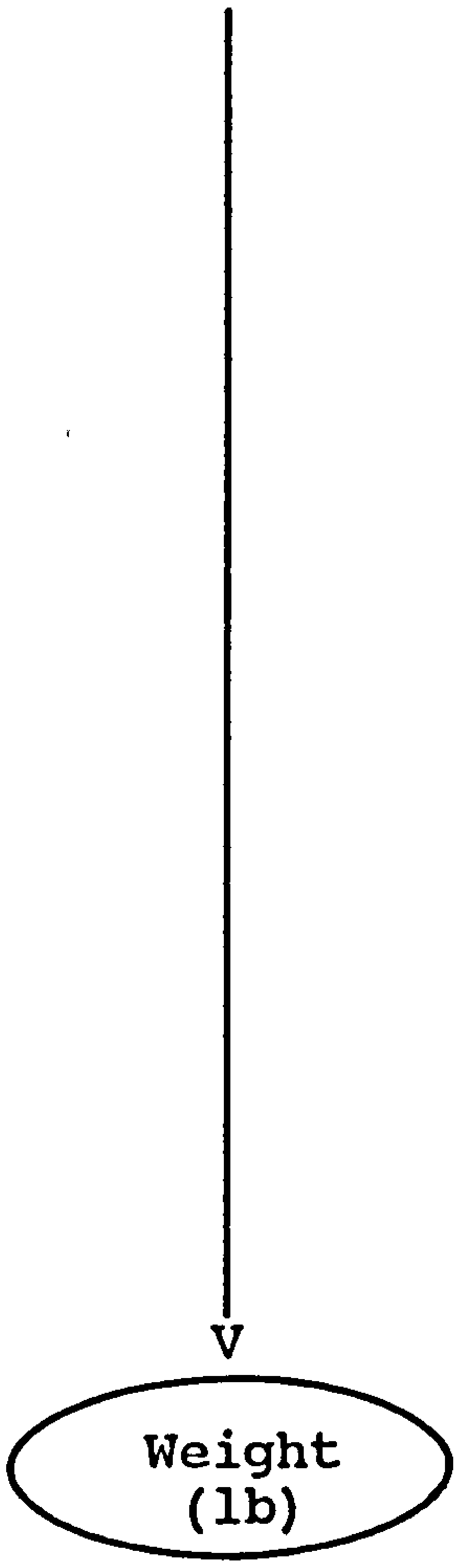
   Dia = seat width (16.5") * seat abreast + Arm rest (2") * (seat abreast + No of Aisle + 1) + Aisle Width * No. of Aisle + Gap(2") * 2+ Thickness (4" to 6" , or 0.02 * Dia_{int} + 1.0).

2. Fuselage Length, L_{fuse} = (No. of passenger / seat abreast) * pitch (29.5" or 34") + No. of Entry Door * 10" + Nose Length(1.5 to 2.0 * Dia) + Tail Length (2.5 to 3.0*Dia).

. Compare with other methods

1. Method 1 [3]

. 1 aisle



Dia=1.25* (16.5" * 2.54" * seat abreast / 100) + 2.3

- 2 aisle

Dia=1.25* (16.5" * 2.54" * seat abreast / 100) + 3.9

- Length= 10 (1.0396 * log (pitch * pax/ seat abreast) + (1.5 to 2) * Dia + (2.5 to 3) * Dia

2. Method [4]

Dia = 2.3 * Number of seat abreast(NB) - 1.1(NB >=5)

Length = 3.76 * Number of Pax / Seat Abreast + 33.2

1. Equation

$W_{Fuse}=(_{1/2}2* Length * Dia * VD)_{1.5}$

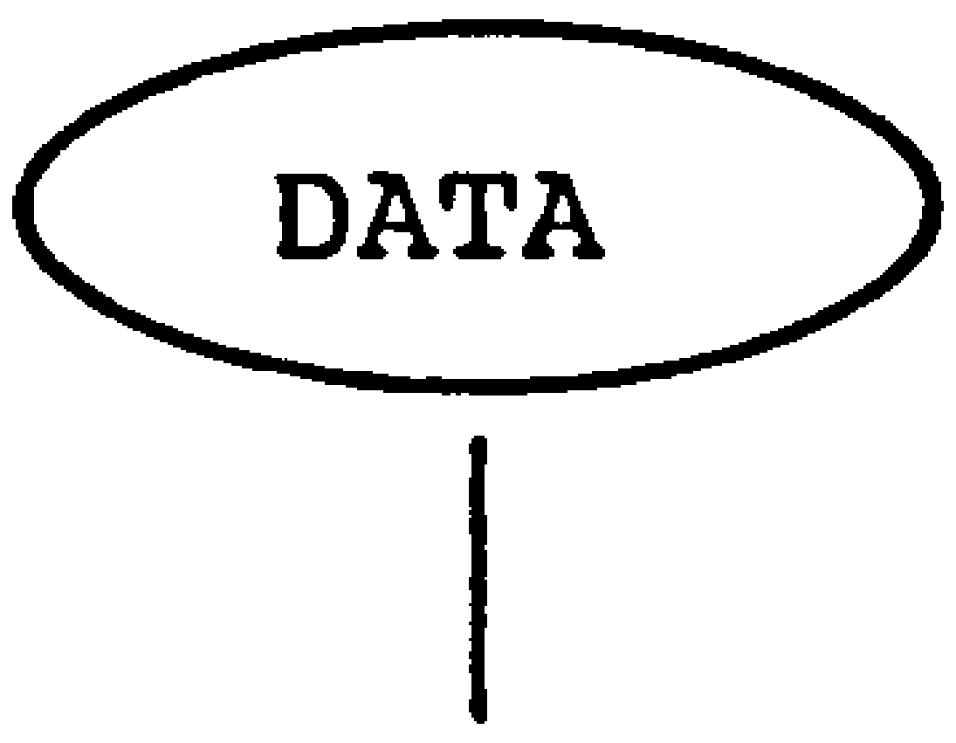
2. With composite

$W_{fuse,comp}=W^{Fuse}*0.8$ (20 % reduced)

I.4 Engine Design (Selection)

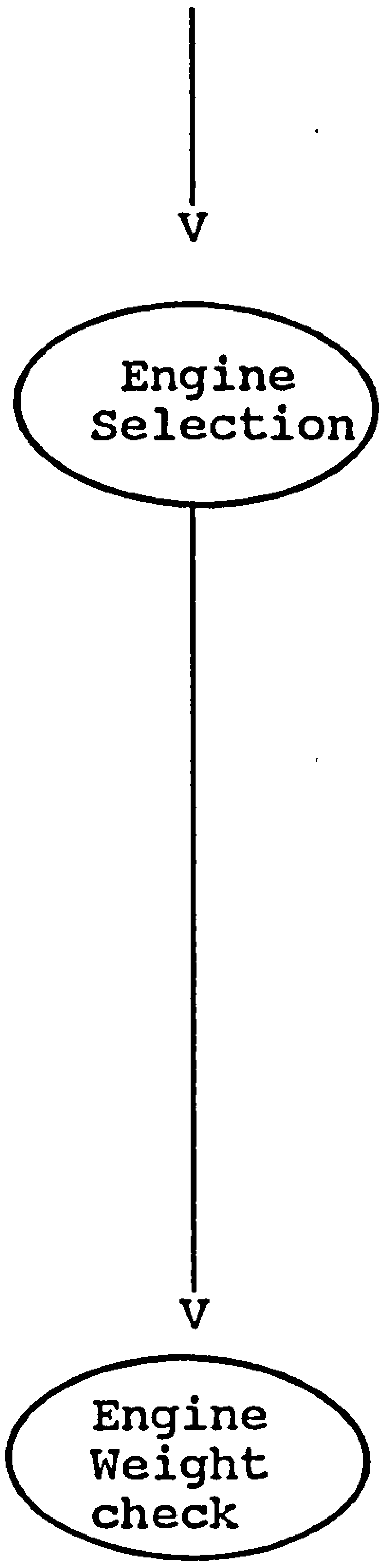
I.4.1 Procedures

PROCESS | **EQUATION AND DATA**



1. From Parametric Study

- Total Take-off Thrust
- By-pass Ratio



- No. of Engines
- Altitude
- Cruise Mach Number
- Specific Fuel Consumption
- Others

1. Each engine Take-off thrust

$$= \frac{\text{Total Take-off Thrust}}{\text{Number of Engines}}$$

2. Selection From Data

To make an allowance of installation loss , the 5 % of pure thrust at the test stand, the 95 % of Engine Thrust from Data should be slightly greater than each engine Take-off thrust.

1. Total engine weight

= Each engine weight * Number of Engines

2. A ratio of the total engine weight to Take-off thrust is checked whether it falls within the following range. [3]

$$.17 =< \frac{\text{Total Engine Weight}}{\text{Take-off Thrust}} =< .25$$

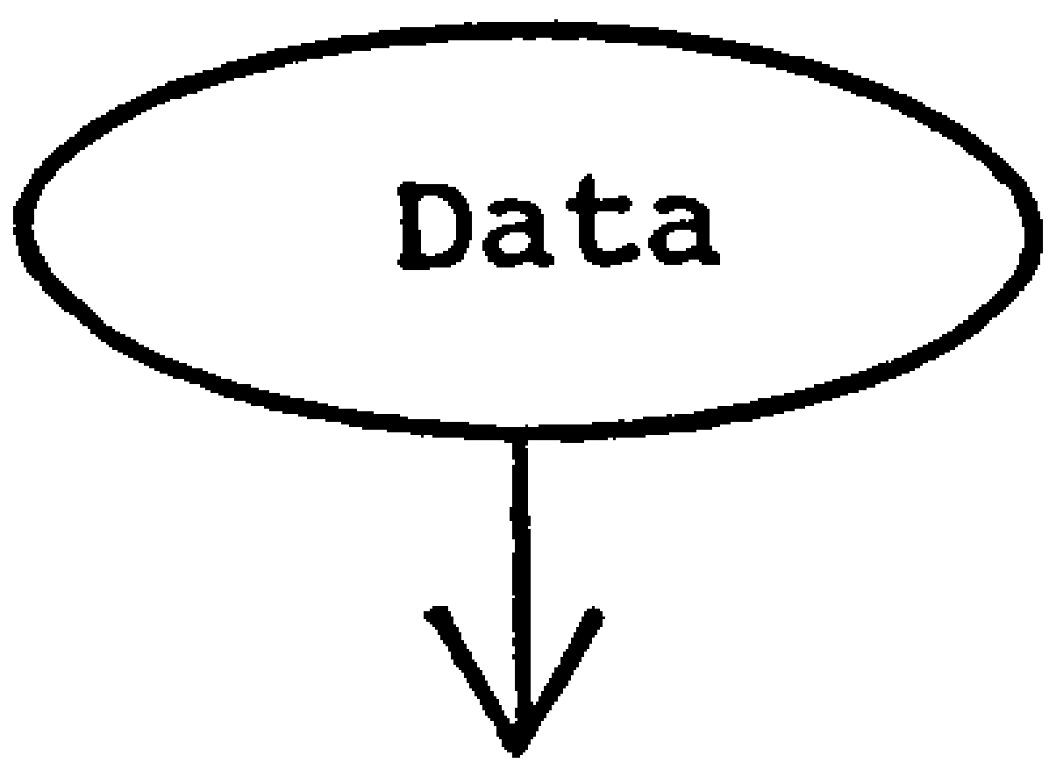
I.5 Tailplane

I.5.1 Input Data and Output Data

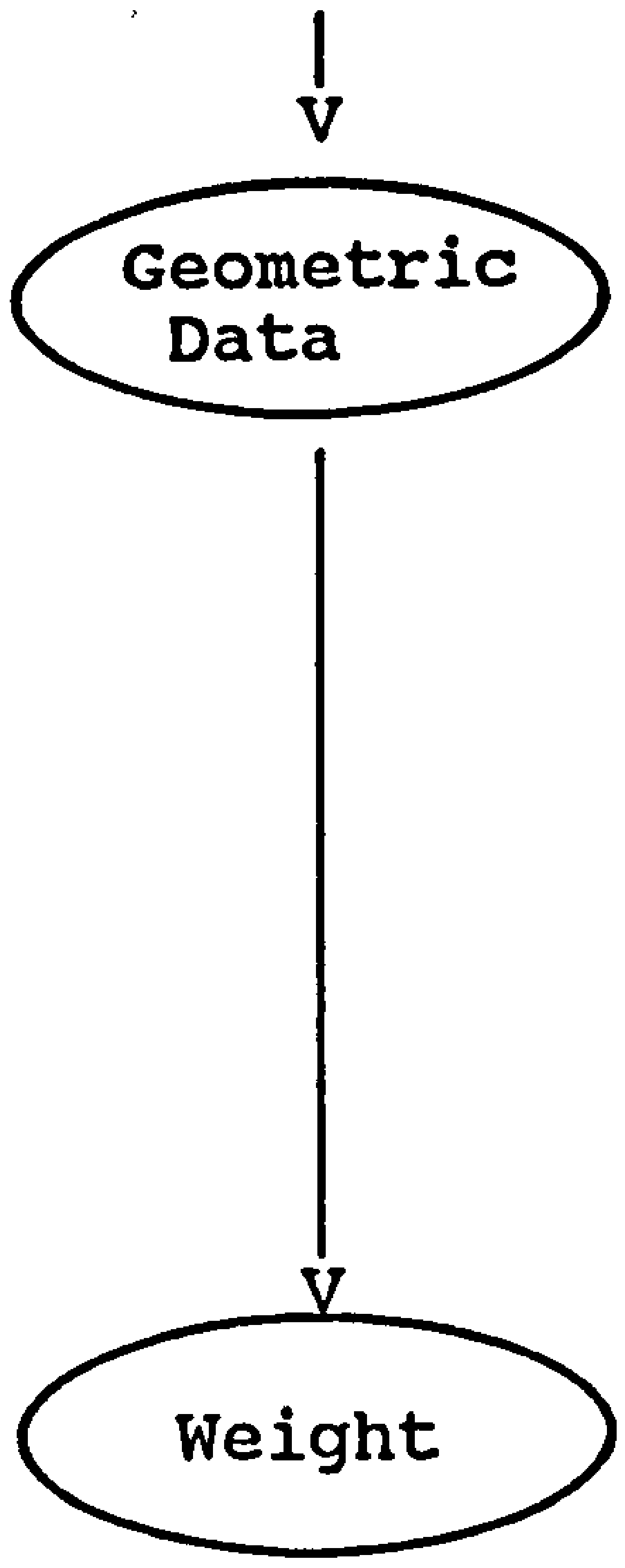
Input	Output	
	Horizontal Stabilizer	Vertical Stabilizer
1. From Parametric Study & Wing Design	$S_{H,T}$	$S_{V,T}$
. Wing Area		
. Wing Chord		
. Wing Span		
. Aspect Ratio	Assume 4	Assume 2
. Swept Angle	$\Lambda_w < \Lambda_{H,T}$	
. Taper Ratio	Assume 0.33	Assume 0.33
. Root Chord		
. Tip Chord		
. Root Thickness		
. Tip Thickness		
2. Others		
. $C_{H,T}$ & $C_{V,T}$	Assume 1.1	Assume 0.08
. $l_{H,T}$ & $l_{V,T}$		
. Airfoil Shape	To select	Symmetric
. Incidence & Twist Angle	0 Assume	0 Assume

I.5.2 Procedures

PROCESS **EQUATION AND DATA**



1. Data from parametric study & wing design



1. For Horizontal and Vertical Stabilizer

   - Planform area
   - Swept Angle
   - Aspect Ratio
   - Taper Ratio
   - Span
   - Dihedral
   - Incidence
   - Volume Coefficient
   - $l_{H,T}$

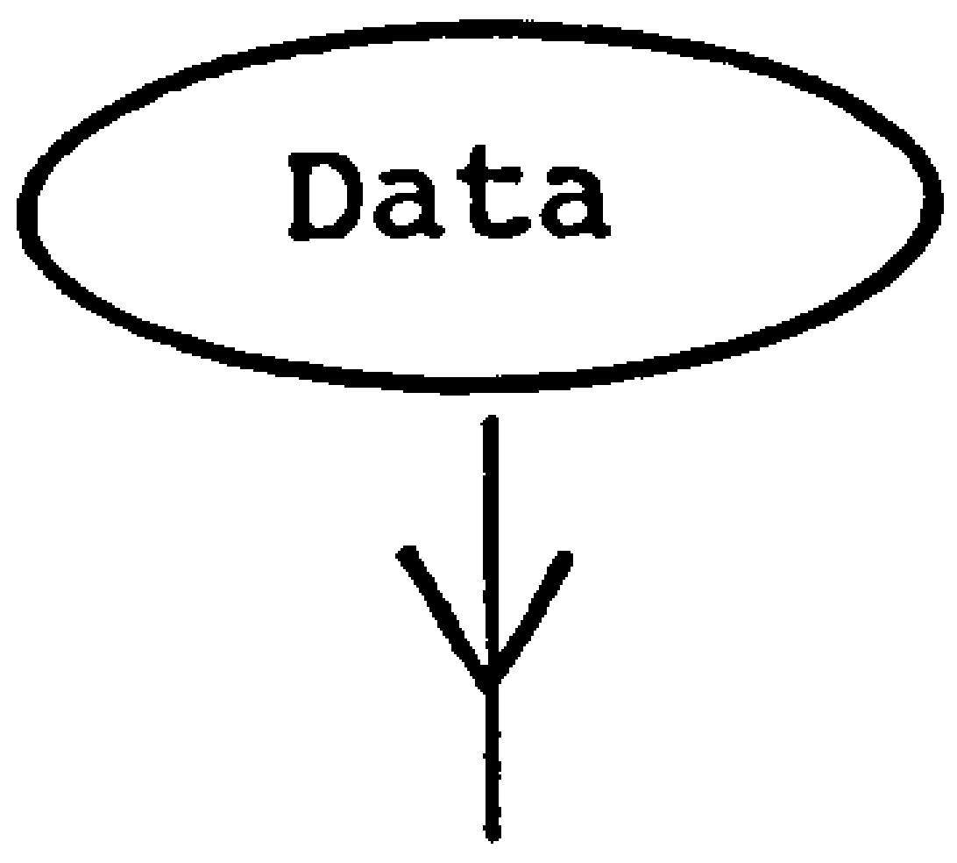
$W_H=K_H*S_H(3.81*\{S_H^{0.2}*V_D\}/\{1,000*(\cos/\backslash_{1/2h})^{1/2})-0.287)$

$W_V=K_V*S_V(3.81*\{S_V^{0.2}*V_D\}/\{1,000*(\cos/\backslash_{1/2h})^{1/2})-0.287)$

I.6 Landing Gear

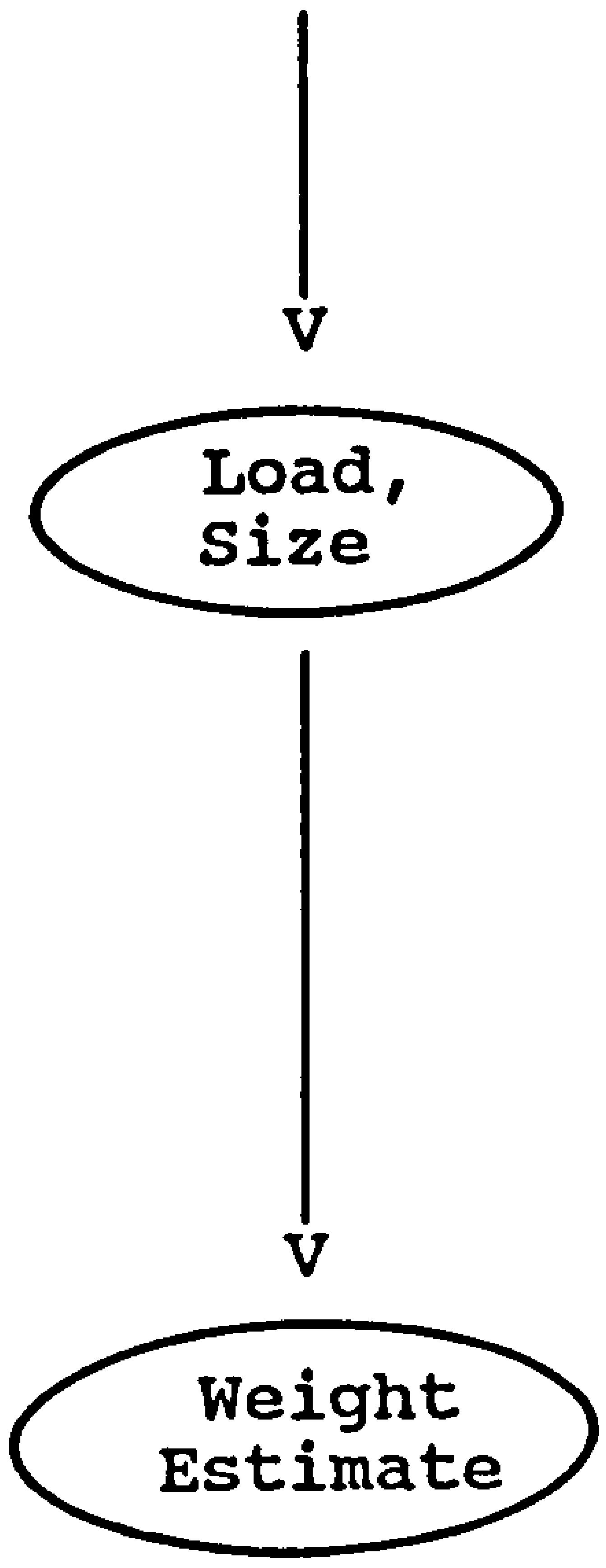
I.6.1 Procedures

PROCESS | EQUATION AND DATA



1. From the Parametric Study, the Fuselage Design

   - Maximum Take-off Weight



- Fuselage Length

This process can be done only after the total layout is known. The total layout was too big to implement for this expert system capacity and was thus omitted.

$$W_{u/c}=62.61(W_{TO}/1000)^{0.84}$$

$$W_{u/c} = 0.038 * W_{TO}$$

I.7 Weight Analysis

I.7.1 Introduction

1. Wing Weight

$$W_{wing} = C1*\left[\frac{b*S}{\text{Cos } /\backslash}\ \frac{(1+2*T.R)}{(3+3*T.R)}\left[\frac{M*N}{S}\right]^{0.3}\left[\frac{V_D}{Tau}\right]^{0.5}\right]^{0.9} \text{kgs}$$

b : Wing Span (m)

S : Wing Area (sq. m)
/\ : Swept angle at quarter chord
T.R : Taper Ratio
W_G : Aircraft Gross Weight (kg)
N : Design Load Factor
V_D : Design Diving speed (m/sec EAS)
Tau : Thickness Chord Ratio at the root.
C1 : 0.028 for long range
0.034 for short range

2. Fuselage Weight

$$W_{fuse} = C2*\left[2* L_{fuse} * Dia_{fuse} *(V_D)^{0.5}\right]^{1.5} \quad \text{(kgs)}$$

C2 : 0.027 for short range
0.022 for long range

L_{fuse} : Fuselage Length
Dia_{fuse} : Fuselage Cross - Section Diameter

3. Tail Unit

$W_{tail} = 0.14 *W_G^{0.83}$ (kgs) Horizontal Tail mounted at the rear fuselage mounted

$= 0.16 *W_G^{0.83}$ Horizontal tail mounted at the rear fuselage mounted

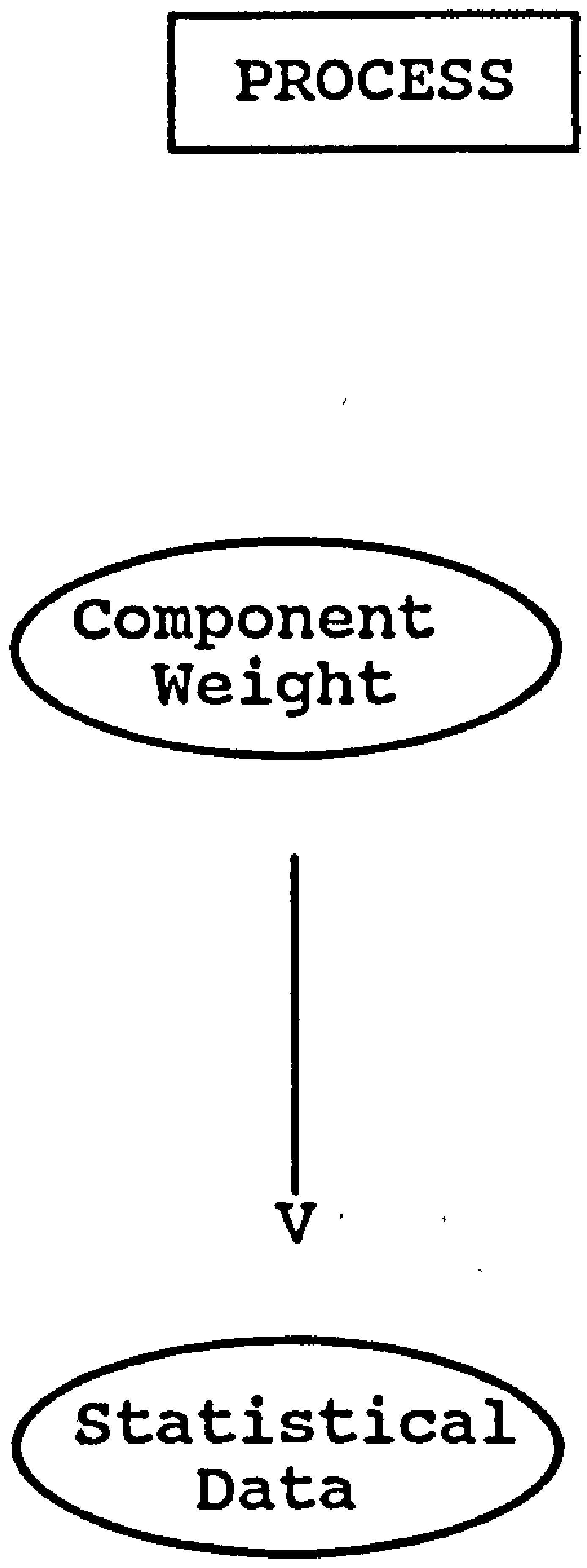
4. Undercarriage

$W_{u/c} = 0.038\ W_{TO}$ (kgs)

5. The Engine Weight is depending upon the existing engine data.

6. Other weight equations were well expressed in reference DES 8336. [2]

I.7.2 Procedure



EQUATION AND DATA

1. Collect weight estimate from each design estimate
2. Equations : Described in Reference [3],[4] and [6].

1. Total Weight

   Structural weight = Wing + Fuselage + Tail + Engine + Miscellaneous

2. Gross Weight

   Gross Weight = Structural Weight + Fuel + Payload

I.8 Cost Analysis

I.8.1 AEA Cost Model

This method calculates the direct operating cost in terms of seat-mile costs and aircraft-mile costs and the descriptions below are abstracted from the reference [27].

1. Utilization : To find out the annual utilization (U), ground manoeuvre time (GT), flight time (CT) and mission block time (MT) must be sought first.

   $$\begin{aligned} GT\ (\text{Hours}) &= (1.152 * R / 14 + 0.05 * R) * 1/60 \\ CT\ (\text{Hours}) &= (R + A) / V \\ BT\ (\text{Hours}) &= CT + BT \\ U &= (3750 / (BT+0.5)) * BT \end{aligned}$$

   R : ranges in nm
   A : 200 nm to allow for the climb & descent
   V : cruise speed in knot as an average flight speed

2. Aircraft Delivery Price (US Dollar) : The aircraft delivery price includes the Manufacturer's Standard Price (MSP), Change Orders (CO), and Capitalized Interest on Progress Payment (CIPP).

   1. Manufacturer's Standard Price (MSP)

      - $AEW = Wg - Wp - Wf - (We * NE)$

        AEW : Airframe Empty Weight
        Wg : Gross Weight
        Wp : Payload Weight
        Wf : Fuel Weight
        We : Engine Weight
        NE : Number of Engines

      - $AP = 27*106 + (9*106 / AEW_{2000nm}) * AEW$

        AEW_{2000nm} is the All Up Weight optimized for 2000 nm.

      - The price of an engine (EP) is regarded to be proportional to the square roots of take-off thrust. For simplicity, the twin-engine turbofan (CF6-80 class) price and four engine turbofan (CFM 56-3 class) price are set at USD 2.6 million and 1.84 million respectively.

        The Take-off thrusts of these engines are 21.77 ton for the CF6-80 and 10.89 ton for the CFM 56-3 respectively.

   - The Electronic Navigational Flight Deck Equipment is fixed at the price of US Dollar 3 Million (the Fiscal Year 1982)

- MSP = AP + NE * EP + 3 * 106
  ADP = 1.09 * MSP

: The factor 1.09 account for the change orders and captilized interest on progress payment.

3. Total Investment is composed of the Aircraft Delivery Price (ADP), Airframe Spares, and Spare Engines Engine Spares.

   1. Airframe Spares = 0.15 * (ADP - NE * EP)

   2. Spare engines and Engine Spares = 0.85 * Bare Engine Unit Price * NE

4. Depreciation (DE)

   DE = Total Investment / (14 * U), US Dollar /Block Hour

5. Interest (INT)

   INT = (0.125/2) * Total Investment * U

6. Insurance (INS)

   INS = 0.005 * ADP / U

7. The crews are assumed to be 2 member flight-deck crews which are on the design trend in modern transport and the cabin crews are 30 per passengers. Costs for flight crew and cabin crew are USD 200 / block hour per flight deck crew and USD 65 / block hour per flight cabin crew.

8. Landing Fees and Navigational Charges

   1. Landing Fees (LF), USD / Block Hour

      LF = 6 * Wg / BT, Wg = metric tons

   2. Navigational Charges (NC)

      NC = (0.5/BT) * Range * (Wg/50)

   3. Ground Handling Charges (GHC)

      GHC = (300 * 50 * Wp) / BT, Wp = metric tons

9. Fuel Cost (FC)

   FC = (0.4 * Wf) / BT

Fuel Price = USD 0.4 / Kg (Fiscal Year 1982)
Wf : Fuel Weight

10. Direct Maintenance Cost (DMC)

1. Airframe Labour (AL), USD / block hour

$AL = ((0.09*Waf + 6.7 - 630/(1.8*Waf+135)) * (1 + 0.59 * (BT-GT))/BT) * (R1*fB)$

$Waf = (Wg-Wp-Wf)*1.02 - (We*NE)$
: Airframe Weight

R1*fB = Direct Labour rate * burden factor on direct Labour + 1 = USD 40 / man hour

2. Airframe Material (AM)

$AM = ((6.24 + 3.08*(BT-GT))/BT) * (ADP - NE*EP)$

3. Direct Engine Maintenance Cost

$EC = NE * (LT+MT)*(BT-GT+1.3)/BT$

LT (Engine Time-dependent Labour)
$= 0.11 * (R1*fB) * C1 * C3 * (1+To)^{0.7}$

$C1 = 1.27 - 0.2 * (BPR)0.2$
$C3 = 0.032 * CS + K$
BPR : Engine By-pass Ratio, 4.5

CS : Number of Engine Compressor Stages including fan,
1. 2 engine A/C : 16
2. 4 engine A/C : 15

K : Function of Number of Shafts, 0.57
PR : Overall Pressure Ratio 2 engine A/C : 28, 4 engine A/C : 25

MT : Engine Material
$= (0.53*(4+To)* C1*(C2+C3) - 1.25) * F$

$C2 = 0.4 * (PR/20)^{1.3} + 0.4$
F : Inflation Factor (1, based on the year 1982)

I.8.2 Direct Operating Costs

1. Total Cost (TC)

TC = DE + INT + INS + 200 / pilot * 2 pilots + (Passenger/30)*65 + LF + NC + GHC + FC + AL + AM + EC

2. Direct Operating Cost per Mile (AC)

AC = (TC / BT) / R

R : Mission Stage Distance

3. Direct Operating Costs per Available Seat Mile (SC)

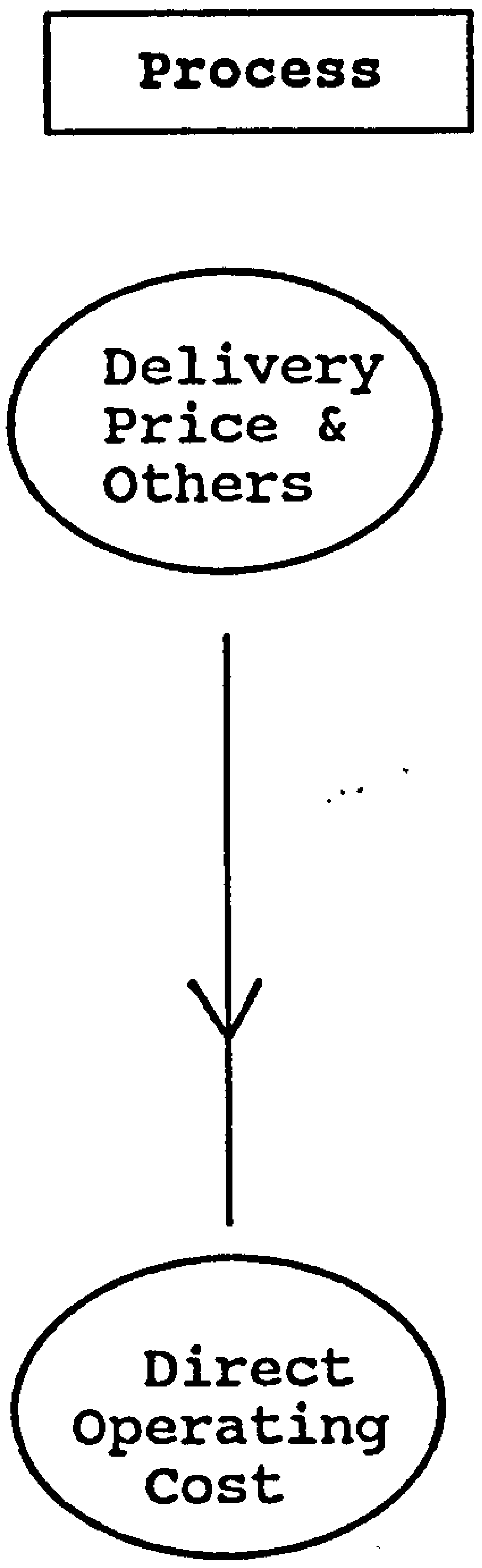
SC = AC / NS

NS : Number of Available seats

I.8.3 Input and Output Data

Input Data	Output Data
1. Specification . Range . Gross Weight . Fuel Weight . Payload . Airframe Empty weight . Take-off Thrust . Number of Engine . Cruising Speed . Number of Pilot 2. Engine Design . Engine Weight . Specific Fuel Consumption 3. Others . Fuel Price . Labour's hourly charge	1. Manufacturer Standard Price (MSP) 2. Total Invest 3. Depreciation 4. Interests / Insurance 5. Landing Fee 6. Navigational Charge 7. Fuel Cost 8. Direct Maintenance Cost 9. Direct Operating Cost

I.8.4 Procedure



Data and Equation

1. Data <--- expressed in Input
2. Equation of output <-- described in Section 4.4.9.2

1. Total Cost
2. Direct Operating Cost/Mile
3. Direct Operating Cost per Available Seat Mile

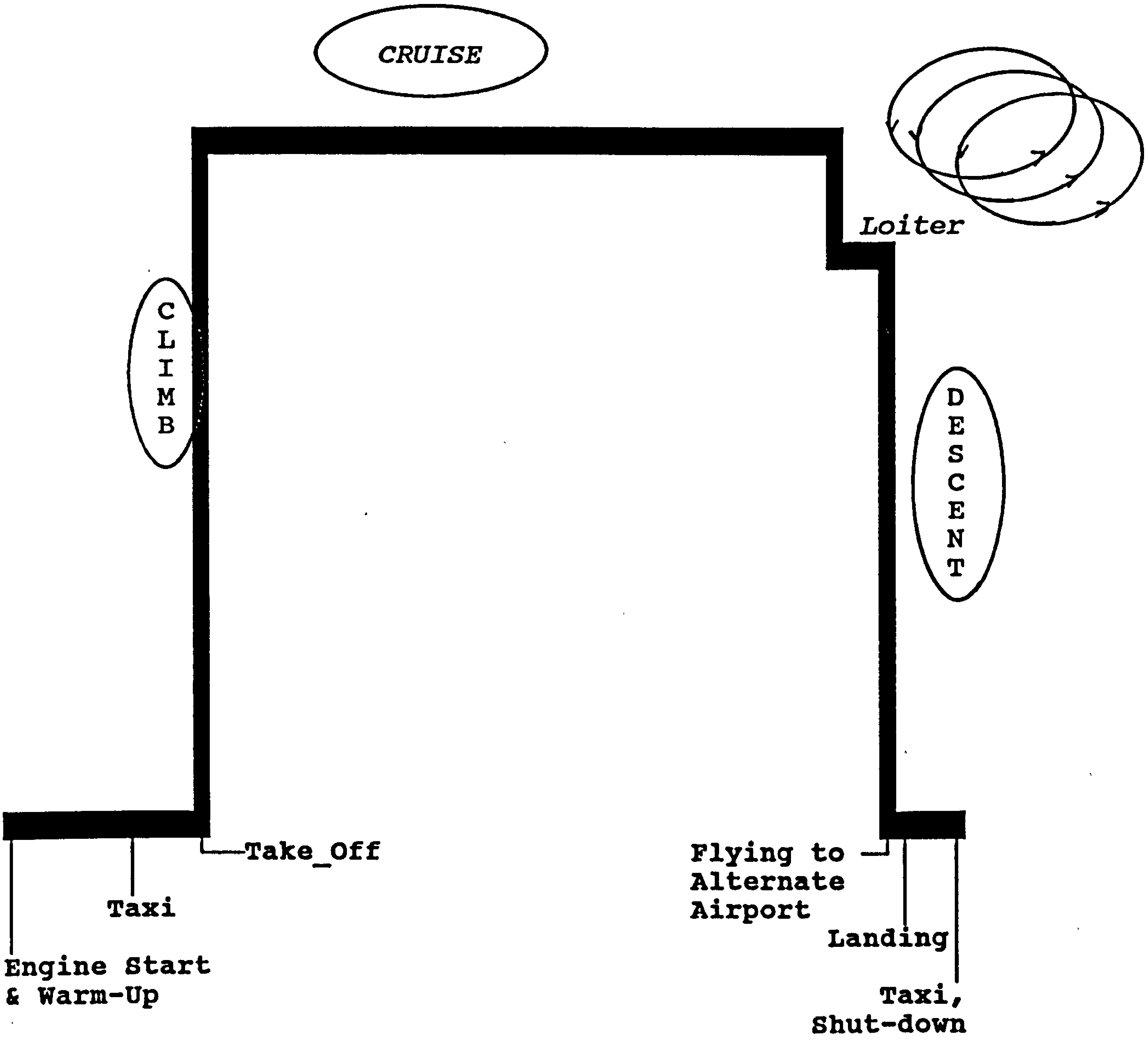


Figure I.1.1/1 The Aircraft Mission Profile

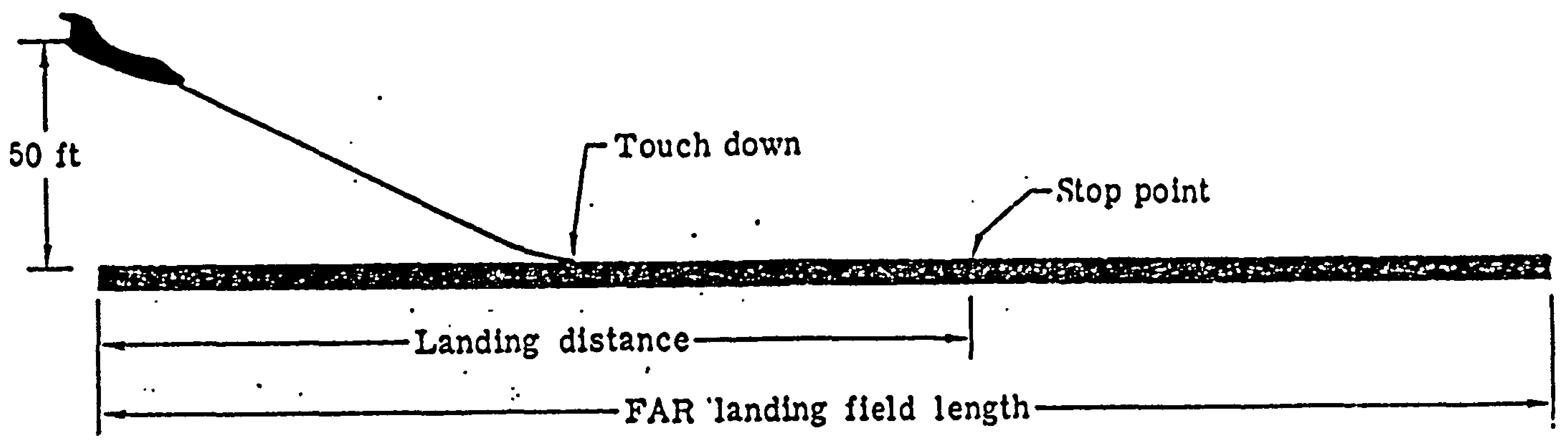


Figure I.1.3/1 The F.A.R. Landing Field Length

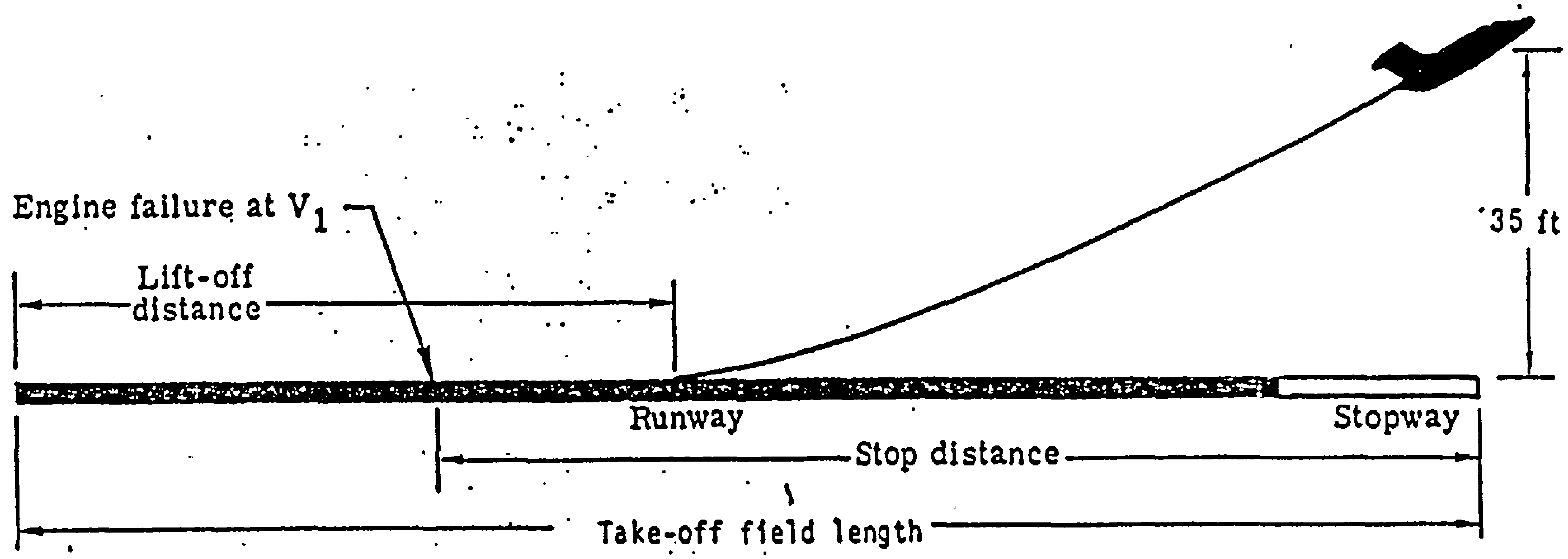


Figure I.1.3/2 The F.A.R. Take-Off Field Length

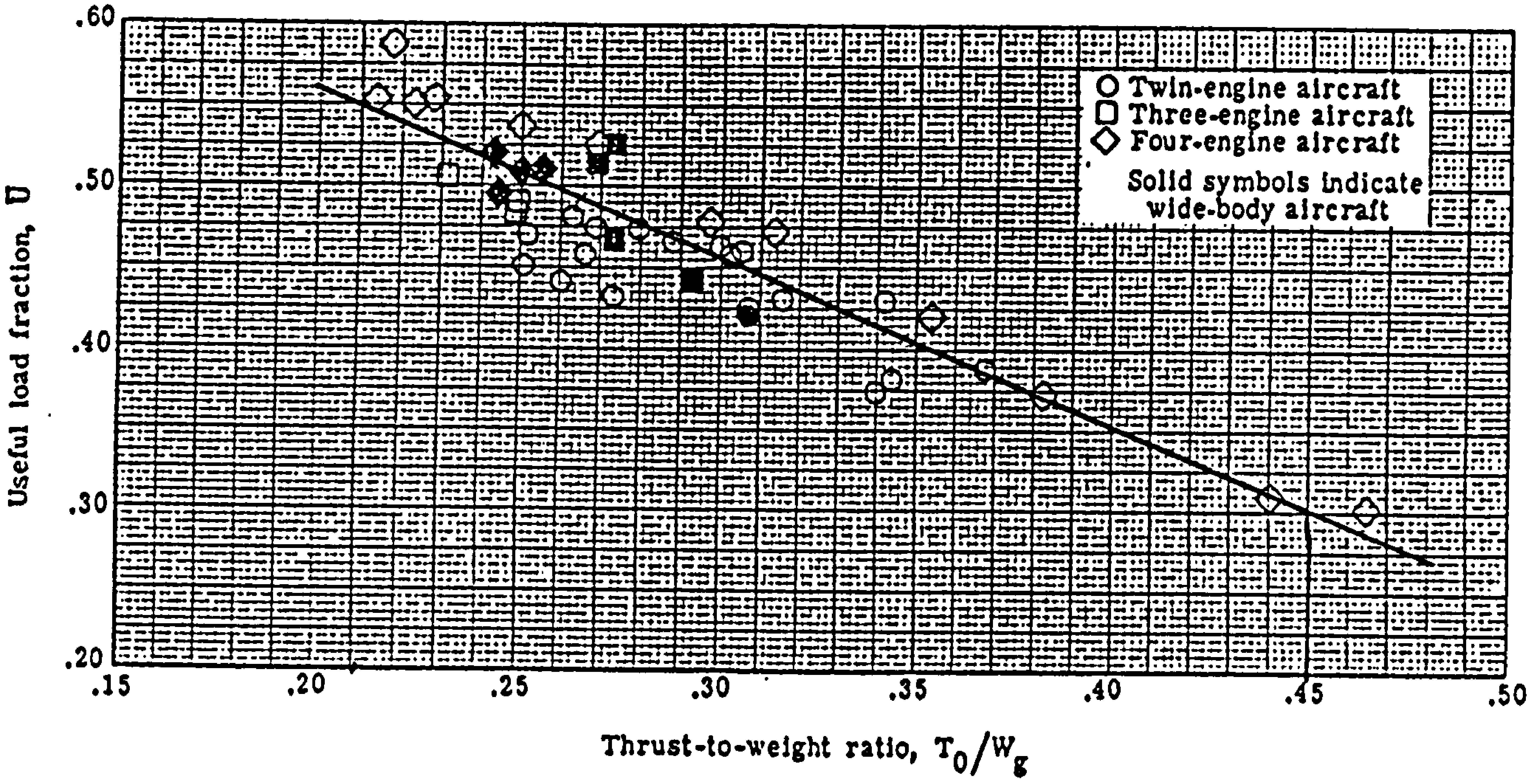


Figure I.1.3/3 The Trend of Useful Load Fraction as Function of Take-Off Thrust to Weight Ratio (p150, [7])

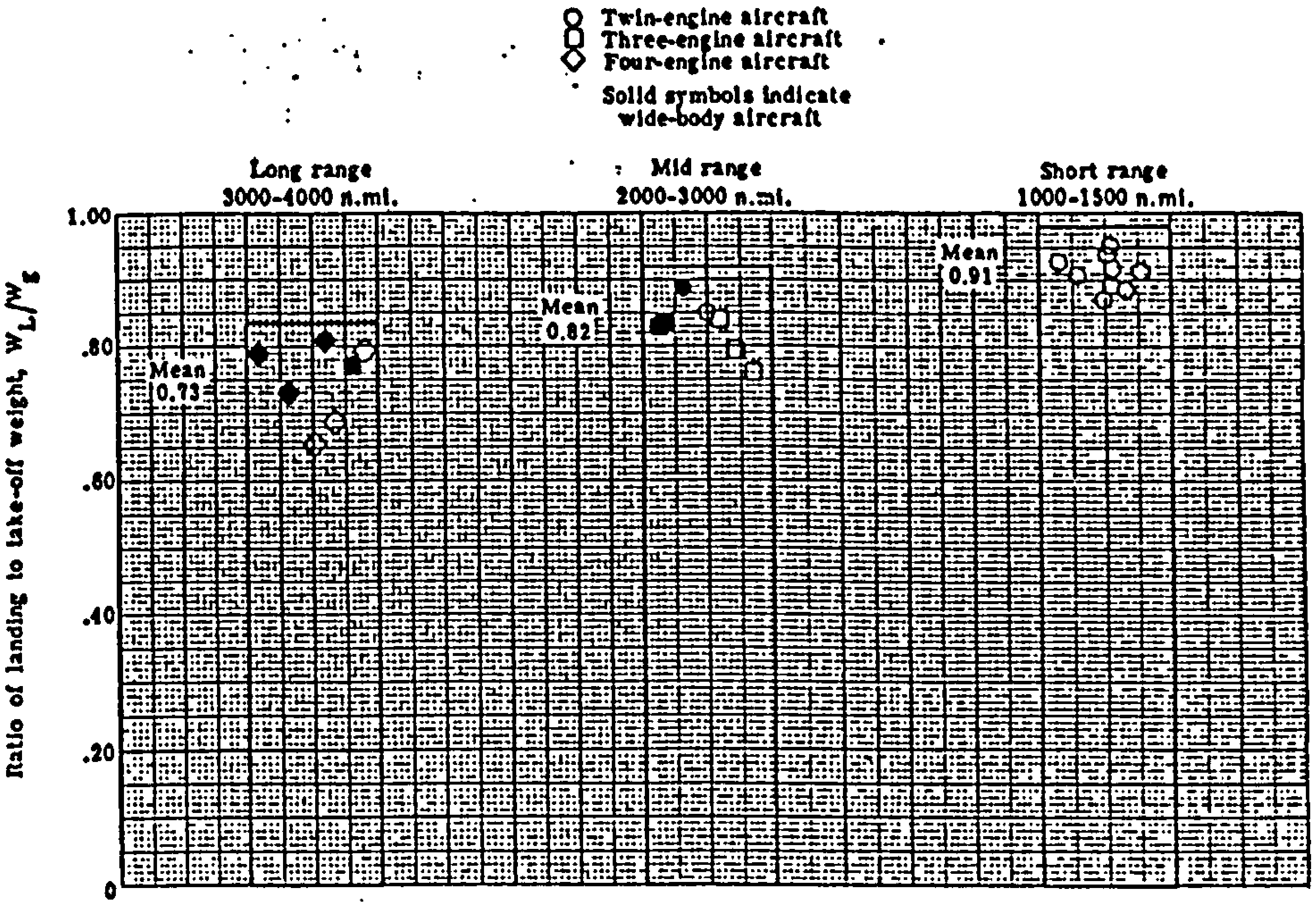


Figure I.1.3/4 The Weight Ratio between Take-Off and Landing (p119, [7])

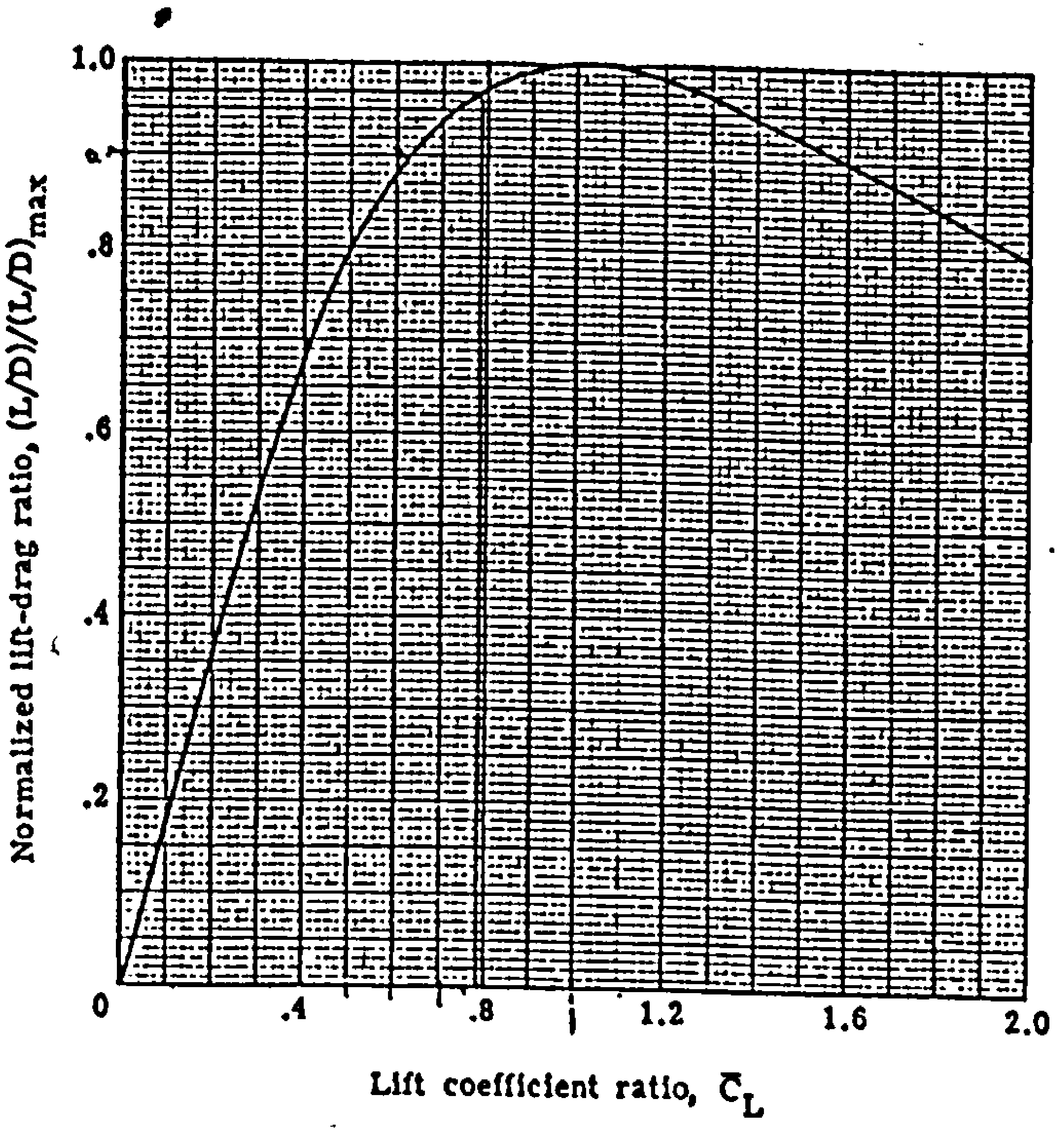


Figure I.1.3/5 The Relationship between Normalized Lift Coefficient Ratio and Normalized Lift to Drag Ratio (p131, [7])

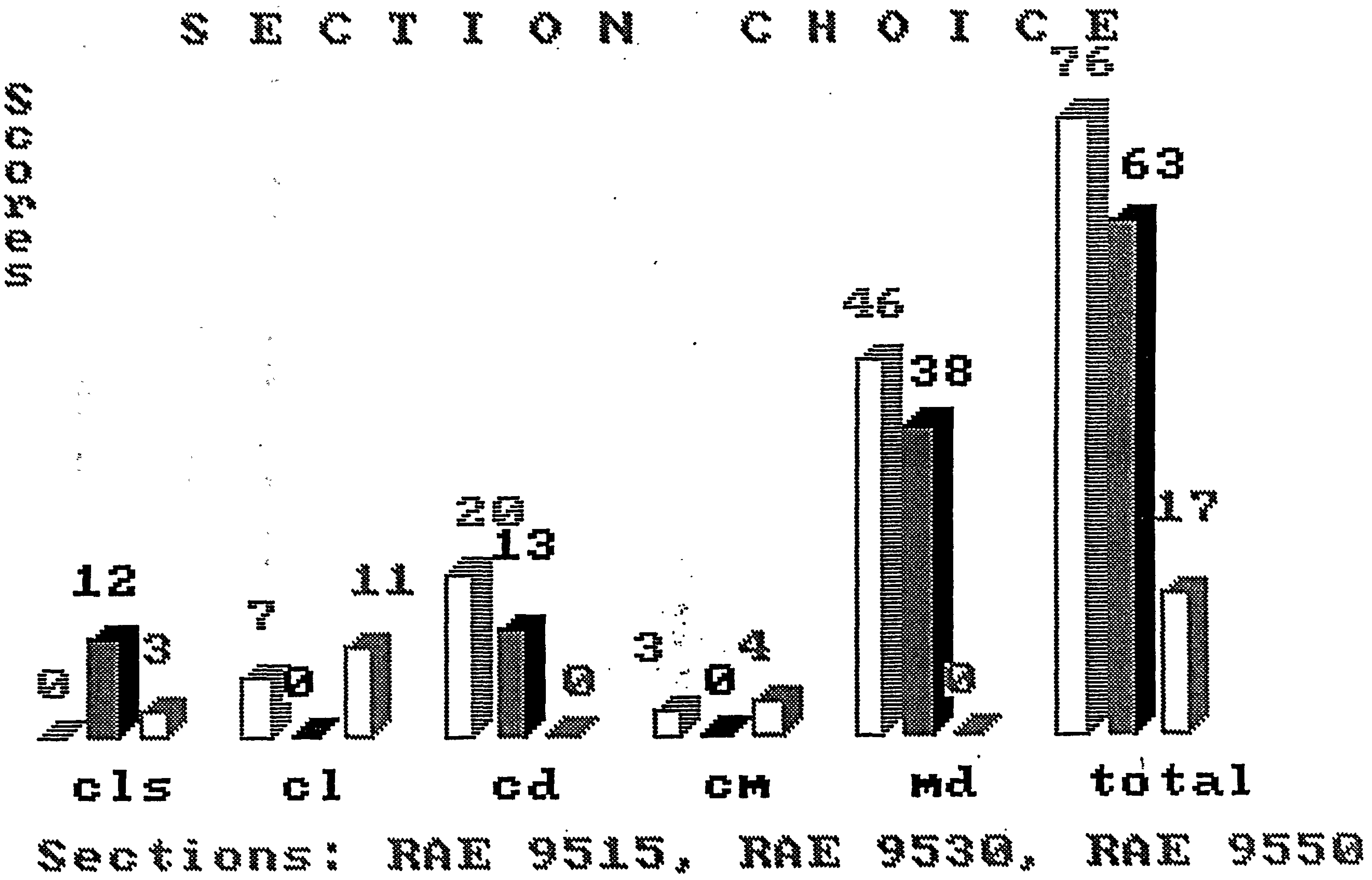


Figure 1.2.3/1 RAE Supercritical Airfoil Selection

APPENDIX II

RULE EXPRESSIONS

Configuration Rule

1. If no condition, then LAYER is VEHICLE and NODE is AIRCRAFT.
2. If no condition, then LAYER is VEHICLE and NODE is SPACECRAFT.
3. If NODE is AIRCRAFT, then LAYER is PURPOSE and NODE is CIVIL.
4. If NODE is AIRCRAFT, then LAYER is PURPOSE and NODE is MILITARY.
5. If NODE is CIVIL, then LAYER is CATEGORY and NODE is LIGHT_AIRCRAFT.
6. If NODE is CIVIL, then LAYER is CATEGORY and NODE is BUSI_EXEC_AIRCRAFT.
7. If NODE is CIVIL, then LAYER is CATEGORY and NODE is TRANSPORT.
8. If NODE is CIVIL, then LAYER is CATEGORY and NODE is CIVIL_CARGO.
9. If NODE is CIVIL, then LAYER is CATEGORY and NODE is CIVIL_ROTORCRAFT.
10. If NODE is MILITARY, then LAYER is CATEGORY and NODE is FIGHTER.
11. If NODE is MILITARY, then LAYER is CATEGORY and NODE is INTERCEPTOR.
12. If NODE is MILITARY, then LAYER is CATEGORY and NODE is TRAINER.
13. If NODE is MILITARY, then LAYER is CATEGORY and NODE is CLOSE_AIR_SUPPORT.
14. If NODE is MILITARY, then LAYER is CATEGORY and NODE is RECONNAISSANCE.
15. If NODE is MILITARY, then LAYER is CATEGORY and NODE is PATROL.

16. If NODE is MILITARY,
then LAYER is CATEGORY and NODE is MILITARY_CARGO.

17. If NODE is MILITARY,
then LAYER is CATEGORY and NODE is MILITARY_ROTORCRAFT.

18. If NODE is MILITARY,
then LAYER is CATEGORY and NODE is BOMBER.

19. If maximum cruising speed in specification is less than 0.9,
then LAYER is SPEED and NODE is SUBSONIC.

20. If maximum cruising speed in specification is less than 1.2 and greater than 0.9,
then LAYER is SPEED and NODE is TRANSONIC.

21. If maximum cruising speed in specification is less than 5.0 and greater than 1.2,
then LAYER is SPEED and NODE is SUPERSONIC.

22. If maximum cruising speed in specification is greater than 5.0,
then LAYER is SPEED and NODE is HYPERSONIC.

23. If no condition, then CONCEPT is CONVENTIONAL.

24. If no condition, then CONCEPT is UNCONVENTIONAL.

25. If landing distance is greater than 3,000 ft and less than 6,000 ft, and takeoff distance is greater than 5,000 ft and less than 12,000 ft,
then LAYER is CONCEPT and NODE is CTOL (Conventional Takeoff and Landing)

26. If landing distance is 0 ft, and takeoff distance is 0 ft,
then LAYER is CONCEPT and NODE is VTOL (Vertical Takeoff and Landing).

27. If landing distance is greater than 0 ft and less 3,000 ft and takeoff distance is greater than 0 ft less than 5,000 ft,
then LAYER is CONCEPT and NODE is STOL (Short Takeoff and Landing).

28. If landing distance is 0 ft and takeoff distance is greater than 0 ft and less than 3,000 ft,
then LAYER is CONCEPT and NODE is STOVL (Short Takeoff and Vertical Landing).

29. If NODE is AIRCRAFT ,
then LAYER is CONFIGURATION_COMPONENT and NODE is FUSELAGE.

30. If NODE is AIRCRAFT,

then LAYER is CONFIGURATION_COMPONENT and NODE is WING.

31. If NODE is AIRCRAFT,
then LAYER is CONFIGURATION_COMPONENT and NODE is ENGINE.

32. If NODE is AIRCRAFT,
then LAYER is CONFIGURATION_COMPONENT and NODE is VERTICAL_TAIL.

33. If NODE is AIRCRAFT,
then LAYER is CONFIGURATION_COMPONENT and NODE is HORIZONTAL_TAIL.

34. If NODE is AIRCRAFT,
then LAYER is CONFIGURATION_COMPONENT and NODE is UNDERCARRIAGE.

35. If NODE is AIRCRAFT,
then LAYER is CONFIGURATION_COMPONENT and NODE is CONFIGURATION_TYPE.

36. If NODE is CONVENTIONAL and NODE is FUSELAGE,
then SUBLAYER is TYPE and NODE is CIRCULAR.

37. If NODE is CONVENTIONAL and NODE is FUSELAGE,
then SUBLAYER is TYPE and NODE is DOUBLE_BUBBLE.

38. If NODE is CONVENTIONAL and NODE is FUSELAGE,
then SUBLAYER is NUMBER and NODE is ONE.

39. If NODE is CONVENTIONAL and NODE is FUSELAGE,
then SUBLAYER is POSITION and NODE is CENTER_LINE.

40. If NODE is CONVENTIONAL and NODE is WING,
then SUBLAYER is TYPE and NODE is BACKWARD_SWEEP.

41. If NODE is CONVENTIONAL and NODE is WING,
then SUBLAYER is NUMBER and NODE is ONE.

42. If NODE is CONVENTIONAL and NODE is WING,
then SUBLAYER is POSITION and NODE is LOW_WING.

43. If NODE is CONVENTIONAL and NODE is WING,
then SUBLAYER is POSITION and NODE is HIGH_WING.

44. If NODE is CONVENTIONAL and NODE is ENGINE and NODE of SPEED is SUBSONIC,
then SUBLAYER is TYPE and NODE is TURBOPROP.

45. If NODE is CONVENTIONAL and NODE is ENGINE and NODE of SPEED is SUBSONIC,
then SUBLAYER is TYPE and NODE is TURBOFAN.

46. If NODE is CONVENTIONAL and NODE is ENGINE and NODE of SPEED is SUBSONIC,
then SUBLAYER is TYPE and NODE is PROPFAN.

47. If NODE is CONVENTIONAL and NODE is ENGINE,
then SUBLAYER is NUMBER and NODE is TWO.

48. If NODE is CONVENTIONAL and NODE is ENGINE
and NODE is TURBOFAN,
then SUBLAYER is NUMBER and NODE is THREE.

49. If NODE is CONVENTIONAL and NODE is ENGINE
and NODE is NOT PROPFAN,
then SUBLAYER is NUMBER and NODE is FOUR.

50. If NODE is CONVENTIONAL and NODE is ENGINE
and NODE is NOT PROPFAN,
then SUBLAYER is POSITION and NODE is
UNDER_WING_MOUNTED.

51. If NODE is CONVENTIONAL and NODE is ENGINE
and NODE is NOT HIGH_WING,
then SUBLAYER is POSITION and NODE is
REAR_FUSELAGE_MOUNTED.

52. If NODE is CONVENTIONAL and NODE is ENGINE
and NODE is PROPFAN,
then SUBLAYER is POSITION and NODE is
REAR_FUSELAGE_MOUNTED.

53. If NODE is CONVENTIONAL and NODE is ENGINE
and NODE is NOT TURBOPROP,
then SUBLAYER is POSITION and NODE is
REAR_FUSELAGE_MOUNTED.

54. If NODE is CONVENTIONAL and NODE is VERTICAL_TAIL,
then SUBLAYER is TYPE and NODE is BACKWARD_SWEEP.

55. If NODE is CONVENTIONAL and NODE is VERTICAL_TAIL,
then SUBLAYER is NUMBER and NODE is ONE.

56. If NODE is CONVENTIONAL and NODE is VERTICAL_TAIL,
then SUBLAYER is POSITION and NODE is
REAR_FUSELAGE_MOUNTED.

57. If NODE is CONVENTIONAL and NODE is
HORIZONTAL_TAIL,
then SUBLAYER is TYPE and NODE is BACKWARD_SWEEP.

58. If NODE is CONVENTIONAL and NODE is
HORIZONTAL_TAIL,
then SUBLAYER is NUMBER and NODE is ONE.

59. If NODE is CONVENTIONAL and NODE is
HORIZONTAL_TAIL,
then SUBLAYER is POSITION and NODE is
VERTICAL_TAIL_MOUNTED.

60. If NODE is CONVENTIONAL and NODE is HORIZONTAL_TAIL
and NODE is HIGH_WING,

then SUBLAYER is POSITION and NODE is
VERTICAL_TAIL_MOUNTED.

61. If NODE is CONVENTIONAL and NODE is HORIZONTAL_TAIL
and NODE of ENGINE POSITION is
REAR_FUSELAGE_MOUNTED,
then SUBLAYER is POSITION and NODE is
VERTICAL_TAIL_MOUNTED.

62. If NODE is CONVENTIONAL and NODE is UNDERCARRIAGE,
then SUBLAYER is TYPE and NODE is RETRACTABLE.

63. If NODE is CONVENTIONAL and NODE is UNDERCARRIAGE,
then SUBLAYER is NUMBER and NODE is THREE.

64. If NODE is CONVENTIONAL and NODE is UNDERCARRIAGE
and number of passenger in specification are
greater than 400,
then SUBLAYER is NUMBER and NODE is FIVE.

65. If NODE is CONVENTIONAL and NODE is UNDERCARRIAGE
and NODE is HIGH_WING,
then SUBLAYER is POSITION and NODE is
NOSE_FUSE_FUSELAGE.

66. If NODE is CONVENTIONAL and NODE is UNDERCARRIAGE
and NODE is LOW_WING and NODE of UNDERCARRIAGE
NUMBER is THREE,
then SUBLAYER is POSITION and NODE is
NOSE_FUSE_WING.

67. If NODE is CONVENTIONAL and NODE is UNDERCARRIAGE
and NODE is LOW_WING and NODE of UNDERCARRIAGE
NUMBER is FIVE,
then SUBLAYER is POSITION and NODE is
NOSE_FUSE_WING_FUSELAGE.

68. If NODE is CONVENTIONAL and NODE is
CONFIGURATION_TYPE
and NODE of WING POSITION is LOW_WING
and NODE of ENGINE TYPE is TURBOFAN
and NODE of ENGINE NUMBER is TWO
and NODE of ENGINE POSITION is UNDER_WING_MOUNTED
and NODE of VERTICAL_TAIL is REAR_FUSELAGE_MOUNTED,

then SUBLAYER is TYPE and NODE is TYPE_1.

69. If NODE is CONVENTIONAL and NODE is
CONFIGURATION_TYPE
and NODE of WING POSITION is LOW_WING
and NODE of ENGINE TYPE is TURBOFAN
and NODE of ENGINE NUMBER is TWO
and NODE of ENGINE POSITION is UNDER_WING_MOUNTED
and NODE of VERTICAL_TAIL POSITION is
VERTICAL_TAIL_MOUNTED,

then SUBLAYER is TYPE and NODE is TYPE_2.

70. If NODE is CONVENTIONAL and NODE is
CONFIGURATION_TYPE
and NODE of WING POSITION is LOW_WING
and NODE of ENGINE TYPE is TURBOFAN
and NODE of ENGINE NUMBER is TWO
and NODE of ENGINE POSITION is
REAR_FUSELAGE_MOUNTED
and NODE of VERTICAL_TAIL is VERTICAL_TAIL_MOUNTED,

then SUBLAYER is TYPE and NODE is TYPE_3.

71. If NODE is CONVENTIONAL and NODE is
CONFIGURATION_TYPE
and NODE of WING POSITION is HIGH_WING
and NODE of ENGINE TYPE is TURBOFAN
and NODE of ENGINE NUMBER is TWO
and NODE of ENGINE POSITION is UNDER_WING_MOUNTED
and NODE of VERTICAL_TAIL is VERTICAL_TAIL_MOUNTED,

then SUBLAYER is TYPE and NODE is TYPE_4.

72. If NODE is CONVENTIONAL and NODE is
CONFIGURATION_TYPE
and NODE of WING POSITION is LOW_WING
and NODE of ENGINE TYPE is TURBOFAN
and NODE of ENGINE NUMBER is THREE
and NODE of ENGINE POSITION is UNDER_WING_MOUNTED
and NODE of VERTICAL_TAIL is VERTICAL_TAIL_MOUNTED,

then SUBLAYER is TYPE and NODE is TYPE_5.

73. If NODE is CONVENTIONAL and NODE is
CONFIGURATION_TYPE
and NODE of WING POSITION is LOW_WING
and NODE of ENGINE TYPE is TURBOFAN
and NODE of ENGINE NUMBER is THREE
and NODE of ENGINE POSITION is
REAR_FUSELAGE_MOUNTED
and NODE of VERTICAL_TAIL is VERTICAL_TAIL_MOUNTED,

then SUBLAYER is TYPE and NODE is TYPE_6.

74. If NODE is CONVENTIONAL and NODE is
CONFIGURATION_TYPE
and NODE of WING POSITION is LOW_WING
and NODE of ENGINE TYPE is TURBOFAN
and NODE of ENGINE NUMBER is THREE
and NODE of ENGINE POSITION is UNDER_WING_MOUNTED
and NODE of VERTICAL_TAIL is REAR_FUSELAGE_MOUNTED,

then SUBLAYER is TYPE and NODE is TYPE_7.

75. If NODE is CONVENTIONAL and NODE is
CONFIGURATION_TYPE
and NODE of WING POSITION is LOW_WING
and NODE of ENGINE TYPE is TURBOFAN
and NODE of ENGINE NUMBER is FOUR
and NODE of ENGINE POSITION is UNDER_WING_MOUNTED

and NODE of VERTICAL_TAIL is VERTICAL_TAIL_MOUNTED,

then SUBLAYER is TYPE and NODE is TYPE_8.

76. If NODE is CONVENTIONAL and NODE is
CONFIGURATION_TYPE
and NODE of WING POSITION is LOW_WING
and NODE of ENGINE TYPE is TURBOFAN
and NODE of ENGINE NUMBER is FOUR
and NODE of ENGINE POSITION is
REAR_FUSELAGE_MOUNTED
and NODE of VERTICAL_TAIL is VERTICAL_TAIL_MOUNTED,

then SUBLAYER is TYPE and NODE is TYPE_9.

77. If NODE is CONVENTIONAL and NODE is
CONFIGURATION_TYPE
and NODE of WING POSITION is HIGH_WING
and NODE of ENGINE TYPE is TURBOFAN
and NODE of ENGINE NUMBER is FOUR
and NODE of ENGINE POSITION is UNDER_WING_MOUNTED
and NODE of VERTICAL_TAIL is VERTICAL_TAIL_MOUNTED,

then SUBLAYER is TYPE and NODE is TYPE_10.

78. If NODE is CONVENTIONAL and NODE is AIRCRAFT,
then LAYER is DESIGN_ACTIVITY and NODE is
PARAMETRIC_STUDY.

79. If NODE is CONVENTIONAL and NODE is AIRCRAFT,
then LAYER is DESIGN_ACTIVITY and NODE is
WIGN_DESIGN.

80. If NODE is CONVENTIONAL and NODE is AIRCRAFT,
then LAYER is DESIGN_ACTIVITY and NODE is
FUSELAGE_DESIGN.

81. If NODE is CONVENTIONAL and NODE is AIRCRAFT,
then LAYER is DESIGN_ACTIVITY and NODE is
ENGINE_DESIGN.

82. If NODE is CONVENTIONAL and NODE is AIRCRAFT,
then LAYER is DESIGN_ACTIVITY and NODE is
VERTICAL_TAIL_DESIGN.

83. If NODE is CONVENTIONAL and NODE is AIRCRAFT,
then LAYER is DESIGN_ACTIVITY and NODE is
HORIZONTAL_TAIL_DESIGN.

84. If NODE is CONVENTIONAL and NODE is AIRCRAFT,
then LAYER is DESIGN_ACTIVITY and NODE is
UNDERCARRIAGE_DESIGN.

85. If NODE is CONVENTIONAL and NODE is AIRCRAFT,
then LAYER is DESIGN_ACTIVITY and NODE is
WEIGHT_ANALYSIS.

86. If NODE is CONVENTIONAL and NODE is AIRCRAFT,
then LAYER is DESIGN_ACTIVITY and NODE is
COST_ANALYSIS.

87. If NODE is CONVENTIONAL and NODE is
PARAMETRIC_STUDY,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
PAYLOAD_RANGE.

88. If NODE is CONVENTIONAL and NODE is
PARAMETRIC_STUDY,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
LANDING_PERFORMANCE.

89. If NODE is CONVENTIONAL and NODE is
PARAMETRIC_STUDY,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
TAKEOFF_PERFORMANCE.

90. If NODE is CONVENTIONAL and NODE is
PARAMETRIC_STUDY,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
CRUISING_PERFORMANCE.

91. If NODE is CONVENTIONAL and NODE is
PARAMETRIC_STUDY,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
SIZE_MATCHING.

92. If NODE is CONVENTIONAL and NODE is WING_DESIGN,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
AIRFOIL_SELECTION.

93. If NODE is CONVENTIONAL and NODE is WING_DESIGN,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
DRAG_RISE_3D.

94. If NODE is CONVENTIONAL and NODE is WING_DESIGN,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
AEROELASTICITY.

95. If NODE is CONVENTIONAL and NODE is WING_DESIGN,
then SUBLAYER is DESIGN_TEMPLATE and NODE is FLAP.

96. If NODE is CONVENTIONAL and NODE is WING_DESIGN,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
TIPSTALL.

97. If NODE is CONVENTIONAL and NODE is WING_DESIGN,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
WING_WEIGHT.

98. If NODE is CONVENTIONAL and NODE is
FUSELAGE_DESIGN,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
TOTAL_LENGTH.

99. If NODE is CONVENTIONAL and NODE is
FUSELAGE_DESIGN,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
DIAMETER.

100. If NODE is CONVENTIONAL and NODE is
FUSELAGE_DESIGN,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
LENGTH_DIA_RATIO.

101. If NODE is CONVENTIONAL and NODE is
FUSELAGE_DESIGN,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
FUSELAGE_WEIGHT.

102. If NODE is CONVENTIONAL and NODE is ENGINE_DESIGN,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
ENGINE_SELECTION.

103. If NODE is CONVENTIONAL and NODE is ENGINE_DESIGN,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
ENGINE_WEIGHT.

104. If NODE is CONVENTIONAL and NODE is
VERT_TAIL_DESIGN,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
PLANFORM.

105. If NODE is CONVENTIONAL and NODE is
VERT_TAIL_DESIGN,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
VERT_TAIL_WEIGHT.

106. If NODE is CONVENTIONAL and NODE is
HORI_TAIL_DESIGN,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
PLANFORM.

107. If NODE is CONVENTIONAL and NODE is
HORI_TAIL_DESIGN,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
HORI_TAIL_WEIGHT.

108. If NODE is CONVENTIONAL and NODE is UNDERCARRIAGE,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
LAYOUT.

109. If NODE is CONVENTIONAL and NODE is UNDERCARRIAGE,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
UNDERCARRIAGE_WEIGHT.

110. If NODE is CONVENTIONAL and NODE is
WEIGHT_ANALYSIS,
then SUBLAYER is DESIGN_TEMPLATE and NODE is
COMPONENT_WEIGHT.

111. If NODE is CONVENTIONAL and NODE is
WEIGHT_ANALYSIS,

then SUBLAYER is DESIGN_TEMPLATE and NODE is TOTAL_WEIGHT.

112. If NODE is CONVENTIONAL and NODE is COST_ANALYSIS, then SUBLAYER is DESIGN_TEMPLATE and NODE is DIRECT_OPERATING_COST.

113. If NODE is CONVENTIONAL and NODE is COST_ANALYSIS, then SUBLAYER is DESIGN_TEMPLATE and NODE is INDIRECT_OPERATING_COST.

114. If NODE is CONVENTIONAL and NODE is COST_ANALYSIS, then SUBLAYER is DESIGN_TEMPLATE and NODE is TOTAL_COST.

SELECTION TYPE

This is expressed as LAYER_TYPE(_LAYER,ONE) or LAYER_TYPE(_LAYER,ALL)

115. SELECTION of LAYER VEHICLE is ONE.
116. SELECTION of LAYER PURPOSE is ONE.
117. SELECTION of LAYER CATEGORY is ONE.
118. SELECTION of LAYER SPEED is ONE.
119. SELECTION of LAYER CONCEPT is ONE.
120. SELECTION of LAYER TAKEOFF_LAND is ONE.
121. SELECTION of LAYER CONFIGURATION_COMPONENT is ALL.
122. SELECTION of LAYER TYPE is ONE.
123. SELECTION of LAYER NUMBER is ONE.
124. SELECTION of LAYER POSITION is ONE.
125. SELECTION of LAYER DESIGN_ACTIVITY is ALL.
126. SELECTION of LAYER DESIGN_TEMPLATE is ALL.

LAYER PROCESS PRIORITY

This is expressed as followed_by(_layer,_next_layer).

127. LAYER VEHICLE must be followed by LAYER PURPOSE.
128. LAYER PURPOSE must be followed by LAYER CATEGORY.
129. LAYER CATEGORY must be followed by LAYER SPEED.
130. LAYER SPEED must be followed by LAYER CONCEPT.

131. LAYER CONCEPT must be followed by LAYER TAKEOFF_LAND.

132. LAYER TAKEOFF_LAND must be followed by LAYER CONFIGURATION_COMPONENT.

133. LAYER CONFIGURATION_COMPONENT must be followed by LAYER DESIGN_ACTIVITY.

134. SUBLAYER TYPE must be followed by SUBLAYER NUMBER.

135. SUBLAYER NUMBER must be followed by SUBLAYER POSITION.

NODE PROCESS PRIORITY

This is expressed as followed_by(_layer,_next_layer).

136. NODE FUSELAGE must be followed by NODE WING.

137. NODE WING must be followed by NODE ENGINE.

138. NODE ENGINE must be followed by NODE VERTICAL_TAIL.

139. NODE VERTICAL_TAIL must be followed by NODE HORIZONTAL_TAIL.

140. NODE HORIZONTAL_TAIL must be followed by NODE UNDERCARRIAGE.

141. NODE UNDERCARRIAGE must be followed by NODE CONFIGURATION_TYPE.

142. NODE PARAMETRIC_STUDY must be followed by NODE WING_DESIGN.

143. NODE WING_DESIGN must be followed by NODE FUSELAGE_DESIGN.

144. NODE FUSELAGE_DESIGN must be followed by NODE ENGINE_DESIGN.

145. NODE ENGINE_DESIGN must be followed by NODE VERT_TAIL_DESIGN.

146. NODE ENGINE_DESIGN must be followed by NODE HORI_TAIL_DESIGN.

147. NODE HORI_TAIL_DESIGN must be followed by NODE UNDERCARRIAGE_DESIGN.

148. NODE UNDERCARRIAGE_DESIGN must be followed by NODE WEIGHT_ANALYSIS.

149. NODE WEIGHT_ANALYSIS must be followed by NODE COST_ANALYSIS.

These are incorporated into SUBLAYER & NODE Clause.

150. NODE PAYLOAD_RANGE must be followed by NODE LANDING_PERFORMANCE.

151. NODE PAYLOAD_RANGE must be followed by NODE TAKEOFF_PERFORMANCE.

152. NODE PAYLOAD_RANGE must be followed by NODE CRUISING_PERFORMANCE.

153. NODE CRUISE_PERFORMANCE must be followed by NODE SIZE_MATCHING.

154. NODE AIRFOIL_SELECTION must be followed by NODE DRAG_RISE_3D.

155. NODE DRAG_RISE_3D must be followed by NODE AEROELASTICITY.

156. NODE DRAG_RISE_3D must be followed by NODE FLAP.

157. NODE AEROELASTICITY must be followed by NODE TIP_STALL.

158. NODE AEROELASTICITY must be followed by NODE WING_WEIGHT.

159. NODE DIAMETER must be followed by NODE TOTAL_LENGTH.

160. NODE TOTAL_LENGTH must be followed by NODE LENGTH_DIA_RATIO.

161. NODE LENGTH_DIA_RATIO must be followed by NODE FUSELAGE_WEIGHT.

162. NODE ENGINE_SELECTION must be followed by NODE ENGINE_WEIGHT.

163. NODE PLANFORM must be followed by NODE VERT_TAIL_WEIGHT.

164. NODE PLANFORM must be followed by NODE HORI_TAIL_WEIGHT.

165. NODE LAYOUT must be followed by NODE UNDERCARRIAGE_WEIGHT.

166. NODE COMPONENT_WEIGHT must be followed by NODE TOTAL_WEIGHT.

167. NODE DIRECT_OPERATING_COST must be followed by NODE INDIRECT_OPERATING_COST.

168. NODE INDIRECT_OPERATING_COST must be followed by NODE TOTAL_COST.

APPENDIX III

KNOWLEDGE BASE IN PROLOG EXPRESSION

Declaration of Layers and Nodes

1. GLOBAL PREDICATES

```
layer_node(STRING,STRING)
layer_type(STRING,STRING)
sublayer_node(STRING,STRING,STRING)
followed_by(STRING,STRING)
selected(STRING,STRING)
selecte3(STRING,STRING,STRING)
```

2. CLAUSES

```
layer_type(vehicle,select_one).
layer_type(purpose,select_one).
layer_type(category,select_one).
layer_type(speed,select_one).
layer_type(concept,select_one).
layer_type(takeoff_land,select_one).
layer_type(configuration_component,select_all).
layer_type(type,select_one).
layer_type(number,select_one).
layer_type(position,select_one).
layer_type(design_activity,select_all).
layer_type(design_template,select_all).

followed_by(vehicle,purpose).
followed_by(purpose,category).
followed_by(category,speed).
followed_by(speed,concept).
followed_by(concept,takeoff_land).
followed_by(takeoff_land,configuration_component).
followed_by(configuration_component,design_activity).
followed_by(design_activity,optimization).

followed_by(fuselage,wing).
followed_by(wing,engine).
followed_by(engine,vertical_tail).
followed_by(vertical_tail,horizontal_tail).
followed_by(horizontal_tail,undercarriage).
followed_by(undercarriage,configuration_type).
followed_by(parametric_study,fuselage_design).
followed_by(fuselage_design,wing_design).
followed_by(wing_design,engine_design).
```

```
followed_by(engine_design,vertical_tail_design).
followed_by(vertical_tail_design,horizontal_tail_design).
followed_by(horizontal_tail_design,undercarriage_design).
followed_by(undercarriage_design,weight_analysis).
followed_by(weight_analysis,cost_analysis).

followed_by(type,number).
followed_by(number,position).

followed_by(payload_range,landing_performance).
followed_by(payload_range,takeoff_performance).
followed_by(payload_range,cruise_performance).
followed_by(landing_performance,size_matching).
followed_by(takeoff_performance,size_matching).
followed_by(cruise_performance,size_matching).

followed_by(airfoil_selection,drag_rise_3d).
followed_by(drag_rise_3d,aeroelasticity).
followed_by(drag_rise_3d,flap).
followed_by(aeroelasticity,tip_stall).
followed_by(aeroelasticity,wing_weight).

followed_by(total_length,fuselage_weight).

followed_by(engine_selection,engine_weight).
followed_by(vertical_tail_planform,vertical_tail_weight).
followed_by(horizontal_tail_planform,
            horizontal_tail_weight).
followed_by(component_weight,total_weight).
followed_by(direct_operating_cost,
            indirect_operating_cost).
followed_by(indirect_operating_cost,
            total_operating_cost).


layer_node(vehicle,aircraft).
layer_node(vehicle,spacecraft).
layer_node(purpose,civil):-selected(vehicle,aircraft).
layer_node(purpose,military):-selected(vehicle,aircraft).
layer_node(category,transport):-selected(purpose,civil).
layer_node(category,light_aircraft):-
           selected(purpose,civil).
layer_node(category,busi_exec_aircraft):-
           selected(purpose,civil).
layer_node(category,civil_cargo):-
           selected(purpose,civil).
layer_node(category,civil_rotorcraft):-
           selected(purpose,civil).
layer_node(category,fighter):-
           selected(purpose,military).
layer_node(category,interceptor):-
           selected(purpose,military).
layer_node(category,trainer):-selected(purpose,military).
layer_node(category,close_air_support):-
           selected(purpose,military).
layer_node(category,reconnaissance):-
           selected(purpose,military).
layer_node(category,patrol):-selected(purpose,military).
```

```
layer_node(category,milit_cargo):-
           selected(purpose,military).
layer_node(category,milit_rotorcraft):-
           selected(purpose,military).
layer_node(category,bomber):-selected(purpose,military).
layer_node(speed,subsonic):-
           user_r(mmax,_Mcr),0<_Mcr,_Mcr<=0.9.
layer_node(speed,transonic):-
           user_r(mmax,_Mcr),0.9<_Mcr,_Mcr<=1.2.
layer_node(speed,supersonic):-
           user_r(mmax,_Mcr),1.2<_Mcr,_Mcr<=5.0.
layer_node(speed,hypersonic):-user_r(mmax,_Mcr),5.0<_Mcr.
layer_node(concept,conventional).
layer_node(concept,unconventional).

layer_node(takeoff_land,ctol):-
           user_r(land_d,_Dist),user_r(t_o_d,_t_o_d),
           3000<_Dist,_Dist<=6000,
           5000<=_t_o_d,_t_o_d<12000.
           layer_node(takeoff_land,stol):-
           user_r(land_d,_Dist),user_r(t_o_d,_t_o_d),
           0<_Dist,_Dist<=3000, 0.0<_t_o_d,_t_o_d<5000.

layer_node(takeoff_land,vtol):-
           user_r(land_d,_Dist),user_r(t_o_d,_t_o_d),
           _Dist=0.0, _t_o_d=0.0.

layer_node(takeoff_land,stovl):-
           user_r(land_d,_Dist),user_r(t_o_d,_t_o_d),
           _Dist=0.0, 0<=_t_o_d,_t_o_d<=5000.

layer_node(configuration_component,fuselage).
layer_node(configuration_component,wing).
layer_node(configuration_component,engine).
layer_node(configuration_component,vertical_tail).
layer_node(configuration_component,horizontal_tail).
layer_node(configuration_component,undercarriage).
layer_node(configuration_component,configuration_type).
layer_node(design_activity,parametric_study).
layer_node(design_activity,fuselage_design).
layer_node(design_activity,wing_design).
layer_node(design_activity,engine_design).
layer_node(design_activity,vertical_tail_design).
layer_node(design_activity,horizontal_tail_design).
layer_node(design_activity,undercarriage_design).
layer_node(design_activity,weight_analysis).
layer_node(design_activity,cost_analysis).

sublayer_node(fuselage,type,circular).
sublayer_node(fuselage,type,double_bubble).
sublayer_node(wing,type,backward_sweep).
sublayer_node(engine,type,turbofan).
sublayer_node(engine,type,turboprop).
sublayer_node(engine,type,propfan):-
              selecte3(wing,position,low_wing).

sublayer_node(vertical_tail,type,backward_swept).
sublayer_node(horizontal_tail,type,back_sweep).
```

```
sublayer_node(undercarriage,type,retractable).
sublayer_node(fuselage,number,one_fuse).
sublayer_node(wing,number,one_wing).
sublayer_node(engine,number,two).
sublayer_node(engine,number,three):-
              selecte3(engine,type,turbofan).
sublayer_node(engine,number,four):-
              not(selecte3(engine,type,propfan)).
sublayer_node(vertical_tail,number,one_vert_tail).
sublayer_node(horizontal_tail,number,one_hori_tail).
sublayer_node(undercarriage,number,three_gear).
sublayer_node(undercarriage,number,five_gear):-
              user_r(pax,_pax),_pax>=400.
sublayer_node(fuselage,position,center_line).
sublayer_node(wing,position,low_wing).
sublayer_node(wing,position,high_wing).

sublayer_node(engine,position,under_wing_mounted):-
              not(selecte3(engine,type,propfan)).
sublayer_node(engine,position,rear_fuselage_mounted):-
              selecte3(engine,type,propfan).
sublayer_node(engine,position,rear_fuselage_mounted):-
              selecte3(wing,position,low_wing).

sublayer_node(vertical_tail,position,
              end_fuselage_mounted).
sublayer_node(horizontal_tail,position,
              vertical_tail_mounted).
sublayer_node(horizontal_tail,position,
              vertical_tail_mounted):-
              selecte3(wing,position,high_wing).
sublayer_node(horizontal_tail,position,
              vertical_tail_mounted):-
              selecte3(engine,position,
              rear_fuselage_mounted).
sublayer_node(horizontal_tail,position,
              after_fuselage_mounted):-
              selecte3(wing,position,low_wing),
              selecte3(engine,position,
              under_wing_mounted).

sublayer_node(undercarriage,position,nose_fuselage_wing):
            - selecte3(undercarriage,number,three_gear),
              selecte3(wing,position,low_wing).
sublayer_node(undercarriage,position,nose_fuse_fuselage):
           -  selecte3(wing,position,high_wing).
sublayer_node(undercarriage,position,
              nose_fuse_wing_fuselage):-
              selecte3(wing,position,low_wing),
              selecte3(undercarriage,number,five_gear).

sublayer_node(parametric_study,design_template,
              payload_range).
sublayer_node(parametric_study,design_template,
              landing_performance).
sublayer_node(parametric_study,design_template,
              takeoff_performance).
sublayer_node(parametric_study,design_template,
```

```
              cruise_performance).
sublayer_node(parametric_study,design_template,
              size_matching).
sublayer_node(wing_design,design_template,
              airfoil_selection).
sublayer_node(wing_design,design_template,drag_rise_3d).
sublayer_node(wing_design,design_template,flap).
sublayer_node(wing_design,design_template,
              aeroelasticity).
sublayer_node(wing_design,design_template,tip_stall).
sublayer_node(wing_design,design_template,wing_weight).

sublayer_node(fuselage_design,design_template,
              total_length).
sublayer_node(fuselage_design,design_template,
              fuselage_weight).
sublayer_node(engine_design,design_template,
              engine_selection).
sublayer_node(engine_design,design_template,
              engine_weight).
sublayer_node(vertical_tail_design,design_template,
              vertical_tail_planform).
sublayer_node(vertical_tail_design,design_template,
              vertical_tail_weight).
sublayer_node(horizontal_tail_design,design_template,
              horizontal_tail_planform).
sublayer_node(horizontal_tail_design,design_template,
              horizontal_tail_weight).

sublayer_node(undercarriage_design,design_template,
              undercarriage_weight).

sublayer_node(weight_analysis,design_template,
              component_weight).
sublayer_node(weight_analysis,design_template,
              total_weight).

sublayer_node(cost_analysis,design_template,
              direct_operating_cost).
sublayer_node(cost_analysis,design_template,
              indirect_operating_cost).
sublayer_node(cost_analysis,design_template,
              total_operating_cost).
```

APPENDIX IV

INFERENCE ENGINE

CONTROL & BACKTRACK

1. CONTROLLER

-. GLOBAL PREDICATES

```
    execute(STRING)
    delete_dependents(STRING)
    delete_dependents(STRING,STRING)
    constraint(string,string)
    execute_node_template(string,string)

    sub_step(STRING,STRING)
    backtrack_to1(STRING,STRING)
    backtrack_to2(STRING,STRING)
    backtrack_to3(STRING,STRING,STRING,STRING)
    execute_layer(STRING)
    select_node(STRING,STRING)
    select_sublayer_node(STRING,STRING,STRING)
    execute_node(STRING,STRING)
    select_sublayer(STRING,STRING)
    execute_sublayer(STRING,STRING)
    execute_sublayer_node(STRING,STRING,STRING)
    check_select(STRINGLIST)
    query1(STRING,STRING,STRINGLIST,STRING)
```

-. CLAUSES

```
execute(optimization).
execute(_layer):-
        backtrack_from(_layer),!,
        execute_layer(_layer).
execute(_layer) :-
        not(backtrack_from(_layer)),
        execute_layer(_layer),
        followed_by(_layer,_next_layer),
        execute(_next_layer).

execute_layer(_layer):-
        findall(NODE,select_node(_layer,NODE),NODES),
```

```
        NODES=[].
execute_layer(_layer):- !,
        findall(_NODE,select_node(_layer,_NODE),_NODES),
        not(_NODES=[ ]),
        concat("Layer ",_layer,_LAY),
        concat(_LAY,"'s ",_LAY1),
        concat(_LAY1," Nodes ",_subject),
        query(design_single,_subject,_NODES,_ans),
        _ans=[_node_name],
        /*design_input(_node_name),*/
        assertz(selected(_layer,_node_name)),
        execute_node(_layer,_node_name),
        not(backtrack_from(_node_name)),
        execute_layer(_layer).

design_input(_node_name):-
       layer_node(Layer,_node_name),
       Layer="design_activity",
       put_screen(_node_name).
design_input(_node_name):-
       layer_node(Layer,_node_name),
       not(Layer="design_activity").

query1(_layer,_subject,_NODES,_node):-
       layer_type(_layer,"select_one"),
       query(single,_subject,_NODES,_ANS),
        _ANS=[_node].
query1(_layer,_,_,_node):-
       layer_type(_layer,"select_all"),
       select_node(_layer,_node).


select_node(_layer,_node):-
        layer_type(_layer,"select_one"),
        layer_node(_layer,_node),
        not(selected(_layer,_)).
select_node(_layer,_node):-
        layer_type(_layer,"select_all"),
        layer_node(_layer,_node),
        followed_by(_node,_),
        not(followed_by(_,_node)),
        not(selected(_layer,_node)).
select_node(_layer,_node):-
        layer_type(_layer,"select_all"),
        layer_node(_layer,_node),
        followed_by(_previous_node,_node),
        not(selected(_layer,_node)),
        selected(_layer,_previous_node).

select_sublayer_node(_main_node,_layer,_node):-
        layer_type(_layer,"select_one"),
        not(selecte3(_main_node,_layer,_)),
        sublayer_node(_main_node,_layer,_node).
select_sublayer_node(_main_node,_layer,_node):-
        layer_type(_layer,"select_all"),
        sublayer_node(_main_node,_layer,_node),
        not(selecte3(_main_node,_layer,_node)),
        findall(_PRE,followed_by(_PRE,_node),_PRES),
```

```
        check_select(_PRES).

select_sublayer(_node,_sublayer):-
        sublayer_node(_node,_sublayer,_),
        followed_by(_sublayer,_),
        not(followed_by(_,_sublayer)),
        not(selecte3(_node,_sublayer,_)).
select_sublayer(_node,_sublayer):-
        sublayer_node(_node,_sublayer,_),
        followed_by(_pre_sublayer,_sublayer),
        selecte3(_node,_pre_sublayer,_),
        not(selecte3(_node,_sublayer,_)).
select_sublayer(_node,_layer):-
        sublayer_node(_node,_layer,_sub),
        not(selecte3(_node,_layer,_sub)),
        not(followed_by(_layer,_)),
        not(followed_by(_,_layer)).

check_select([]):-!.
check_select([H|T]):-!,selecte3(_,_,H), check_select(T).

execute_node(_layer,_node):-
        layer_type(_layer,"select_one"),
        execute_node_template(_layer,_node),
        ask(qn,backtrack,"",ANS),
        backtrack_to1(_node,ANS).
execute_node(_,_node):-
        findall(_sublayer,select_sublayer(_node,_sublayer
        ),SUBS),
        SUBS=[].
execute_node(_layer,_node):-
        select_sublayer(_node,_sublayer),
        execute_sublayer(_node,_sublayer),
        execute_node(_layer,_node).

execute_sublayer(_node,_layer):-
        findall(_N,select_sublayer_node(_node,_layer,_N),
        NODES),
        NODES=[].

execute_sublayer(_node,_layer):-
        findall(_N,select_sublayer_node(_node,_layer,_N),
        _NODES),
        not(_NODES=[ ]),
        concat(_node,"'s ",_nod1),
        concat(_nod1, _layer ,_nod2),
        concat(_nod2," Nodes ",_subject),
        query(design_single,_subject,_NODES,_ANS),
        _ANS=[_node_name],
        assertz(selecte3(_node,_layer,_node_name)),
        execute_sublayer_node(_node,_layer,_node_name),
        execute_sublayer(_node,_layer).


execute_sublayer_node(_node,_layer,_node_name):-
        execute_node_template(_layer,_node_name),
        evaluate(_layer,_node_name),
        ask(qn,backtrack,"",ANS),
```

```
        backtrack_to1(_node,ANS).
```

BACKTRACK MECHANISM

```
backtrack_to1(_node,"Y"):-!,
        findall(LAYER,selected(LAYER,_),LAYERS),
        no_mult(LAYERS,_layers),
        query(single,"Select Layers to Backtrack
to",_layers,_ANS),
        _ANS=[_answer],
        assertz(backtrack_from(_node)),
        backtrack_to2(_node,_answer).
backtrack_to1(_,_).

backtrack_to2(_node,_answer):-
        layer_type(_answer,"select_one"),
        layer_node(LAYER,_node),
        assertz(backtrack_from(LAYER)),
        delete_dependents(_answer),!,
        execute(_answer),
        retract(backtrack_from(LAYER)).
backtrack_to2(_node,_answer):-
        layer_type(_answer,"select_all"),
        layer_node(LAYER,_node),
        findall(NODE,selected(_answer,NODE),_NODES),
        concat("Backtrack Layer ",_answer,_LL),
        concat(_LL,"'s NODES",_LLL),
        query(single,_LLL,_NODES,_ans1),
        _ans1=[_nodename],

findall(_sub1,selecte3(_nodename,_sub1,_),_sub11),
        no_mult(_sub11,_sub12),
        concat(_nodename,"'s sublayers for backtrack
               ",_SS),
        query(single,_SS,_sub12,_su),
        _su=[_sub],
        backtrack_to3(LAYER,_answer,_nodename,_sub).
backtrack_to2(_,_):-retractall(backtrack_from(_)).
backtrack_to2(_,_).

backtrack_to3(LAYER,_answer,_nodename,_sub):-
        not(_sub=design_template),
        assertz(backtrack_from(LAYER)),
        delete_dependents(_nodename,_sub),
        execute(_answer),
```

```
        retract(backtrack_from(LAYER)).

backtrack_to3(LAYER,_answer,_nodename,_sub):-
        _sub=design_template,
        assertz(backtrack_from(LAYER)),
        findall(NODE,selecte3(_nodename,_,NODE),NODES),
        query(single,"Backtrack to Design_template"
        ,NODES,_ANS),
        _ANS=[_node],
        delete_dependents(_node),
        execute_layer(_answer),
        retract(backtrack_from(LAYER)).
```

2. PROBLEM CHECKING

-. PREDICATES

```
   likely_source(STRING, STRING)
   possible_problem(STRING, STRING)
   problem_with(STRING, STRING)
   design_input(STRING)
```

-. CLAUSES

```
problem_with(landing_performance,"Not satisfied: Wing
        loading Check"):-
        selecte3(_,_,landing_performance),
        findall(_A,dataland(_,_,_A,_),_As),
        _As=[].

problem_with(takeoff_performance,"Not satisfied: Wing
        loading Check"):-
        selecte3(_,_,takeoff_performance),
        findall(_A,datato(_,_,_A,_),_As),
        _As=[].

problem_with(cruise_performance,"Not satisfied: Wing
        loading Check"):-
        selecte3(_,_,cruise_performance),
        findall(_A1,datacr(_,_,_A1,_,_),_A1s),
        _A1s=[].

problem_with(size_matching,"Not satisfied: Wing loading
        Check"):-
        selecte3(_,_,size_matching),
        findall(_A1,datamat(_,_A1,_,_,_,_,_),_A1s),
        _A1s=[].
```

```
problem_with(sweep_angle_range,"Not satisfied:
        Aeroelastic check") :-
        selecte3(_,_,aeroelastic),
        findall(_A, angles(aeroelastic_angles,_A), _As),
        _As = [].

problem_with(sweep_angle_range,"Not satisfied: Flap
        check") :-
        selecte3(_,_,flap),
        findall(_A, angles(flap_angles,_A), _As),
        _As = [].

problem_with(sweep_angle_range,"Not satisfied: Tip stall
     check") :-
     selecte3(_,_,tipstall),
     findall(_A, angles(tip_stall_angles,_A), _As),
     _As = [].

problem_with(sweep_angle_range,"Sweep angle conflict:
     Flap and aeroelastic") :-
     selecte3(_,_,flap),
     selecte3(_,_,aeroelastic),
     findall(_A1, angles(flap_angles,_A1), _A1s),
     findall(_A2, angles(aeroelastic_angles,_A2), _A2s),
     intersection(_A1s, _A2s, _As),
     _As = [].

problem_with(sweep_angle_range,"Sweep angle conflict:
     Flap and tip stall") :-
     selecte3(_,_,flap),
     selecte3(_,_,tip_stall),
     findall(_A1, angles(flap_angles,_A1), _A1s),
     findall(_A2, angles(tip_stall_angles,_A2), _A2s),
     intersection(_A1s, _A2s, _As),
     _As = [].

problem_with(sweep_angle_range,"Sweep angle conflict: Tip
     stall and aeroelastic") :-
     selecte3(_,_,aeroelastic),
     selecte3(_,_,tip_stall),
     findall(_A1, angles(aeroelastic_angles,_A1), _A1s),
     findall(_A2, angles(tip_stall_angles,_A2), _A2s),
     intersection(_A1s, _A2s, _As),
     _As = [].

problem_with(high_speed_requirement,"Not satisfied: High
     speed requirement") :-
     not(high_speed).

problem_with(thickness_chord_ratio,"Not satisfied:
     Thickness/chord ratio") :-
     findall(_A, angles(drag_rise_3d_angles,_A), _As),
     _As = [].

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)).
```

3. CHECKING IMPLEMENTATION

-. PREDICATES

```
   backtrack(STRING,STRINGLIST,STRING)
   backtrack_deletion(STRINGLIST,STRING,STRING)
   best_guess_at_source(STRING,STRING,STRING)
   best_resumption_point(STRING,STRING,STRING)
   dependent_step(STRING,STRING)
   evaluate(STRING,STRING)
   evaluate_step(STRING,STRING)
   evaluation(STRING,STRINGLIST)
   evaluation1(STRING,STRING)
   expert_backtracking_choice(STRING,STRINGLIST,STRING)
   possible_source(STRING,STRING)
```

-. CLAUSES

```
backtrack(_step, [], _step) :- !.
backtrack(_step, _problems, _resumption_point) :-
          findall(_s,expert_backtracking_choice(_step,_pr
          oblems,_s),_sources),
          delete_dependents(_step),
          no_mult(_sources,_sources1),  !,
          query(mult, backtrack_design_steps, _sources1,
           _backtrackpoints),
backtrack_deletion(_backtrackpoints,_step,
           _resumption_point).


backtrack_deletion([_backtrackpoint | _bps], _stpin,
           _stpout) :- !,
          best_resumption_point(_backtrackpoint, _stpin,
          _highstp),
          delete_dependents(_backtrackpoint),
          backtrack_deletion(_bps, _highstp, _stpout).
```

```
backtrack_deletion([], _step, _step) :- !.

best_guess_at_source(_problem, _step, _source) :-
          likely_source(_problem, _source),
          possible_source(_step, _source).

best_guess_at_source(_problem, _step, _source) :-
          not(likely_source(_problem,_)),
          possible_source(_step, _source).


best_resumption_point(_lev1,_lev2,_lev1):-
dependent_step(_lev1,_lev2), !.
best_resumption_point(_, _lev2, _lev2) :- !.


delete_dependents(_step) :-
          dependent_step(_step,_s),
          layer_node(_s,_n),
          retractall(selected(_s,_)),
          retractall(selecte3(_n,_,_)),fail.
delete_dependents(_step) :-
          dependent_step(_step,_s),
          retractall(selected(_,_s)),
          retractall(selecte3(_s,_,_)),fail.
delete_dependents(_nod):-
          selecte3(_node,_,_nod),
          retractall(selected(_,_node)),
          dependent_step(_nod,_nn),
          retractall(selecte3(_node,_,_nn)),
          sub_step(_node,_n),
          delete_dependents(_n),fail.
delete_dependents(_).

delete_dependents(_node,_sub):-
          retract(selected(_,_node)),
          sub_step(_node,_n),
          delete_dependents(_n),
          dependent_step(_sub,_s),
          retractall(selecte3(_node,_s,_)),fail.
delete_dependents(_,_).
```

dependent_step(S1,S2) says that S2 is dependent on values from S1 because it is a sub_goal of S1 or shares values with S1 and is always executed after S1.

```
dependent_step(S,S).
dependent_step(S1,S2) :-
          sub_step(S1,S2).
```

```
dependent_step(S1,S2) :-
          layer_node(L1,S1),
          layer_node(L2,S2),
          sub_step(L1,L2).

sub_step(_step,_descendent):-
          followed_by(_step,_descendent).
sub_step(_step,_descendent):-
          followed_by(_step,_s),
          sub_step(_s,_descendent).



evaluate("design_template",_step):-
          evaluate_step(_step,_next_step),!,
          _step=_next_step.
evaluate("design_template",_step):-
          evaluate_step(_step,_next_step),!,
          not(_step=_next_step),
          selecte3(_node,_,_next_step),
          delete_dependents(_next_step),
          execute_sublayer(_node,"design_template").
evaluate(_,_).

evaluate_step(_step, _next_step):-
          evaluation(_step, _problems),
          backtrack(_step, _problems, _next_step).


evaluation(_step, _problems) :-
          concat("Checking Result: ",_step,MSG),
          put_msg(MSG),!,
          only_checks_for(_step, _),
          findall(_p, evaluation1(_step, _p), _problems).
evaluation(_step, _problems) :-
          ask(qn,design_step_problem,"",ANS),
          ANS <> "Y",
          findall(_p, possible_problem(_step, _p),
          _options),
          query(mult, problem_name, _options, _problems),
          !.
evaluation(_,[]) :- !.

evaluation1(_s, _p) :-
          only_checks_for(_s, _c),
          problem_with(_c, _p),
          put_error(_p).

expert_backtracking_choice(_step,_ps,_s) :-
          member(_p,_ps),
          best_guess_at_source(_p, _step, _s).

possible_source(_step, _cause) :-
          selecte3(_,_,_step),
          selecte3(_,_,_cause),
          dependent_step(_cause, _step).
```

APPENDIX V

PROLOG EXPRESSIONS OF AIRCRAFT DESIGN ANALYSIS

```
/*******************************************************
                  V D E S A I N
*******************************************************/

project "VEHICLE"
code =3500
include "VGLOBDEF.PRO"

/********
GLOBAL PREDICATES
     execute_template(STRING, STRING)
     speed_of_sound(REAL,REAL)
**********/

/*DOMAINS
     paa=pa(REAL,REAL)
     palist=paa* */

PREDICATES
     nondeterm find(STRING)
     additional_load_factor(REAL,REAL)
     aeroelastic(STRING,REAL,REAL,REAL,REAL)
     control(STRING)
     d_to_r(REAL,REAL)
     line_eq(REALLIST,REAL)
     range1(STRING,REAL,REAL)
     score_sections
     vangle(STRING,REAL)
     nondeterm solve(REAL,REAL,REAL,REAL,REAL)
     nondeterm rrange(real,real)
     nondeterm select(string,drawing)
      nondeterm intersect( real, real, real, real, real,
                           real,real,real)
     select1(drawing,point)
     findpoints
     intersect1
     intersect2
     intersect3
     intersect4
     gpoint(string,reallist,point)
     process(INTEGER,STRING)


CLAUSES

/*=======================================================
  This is the execution of Nodes in design activity
=======================================================*/

execute_node_template(_layer,T) :-
  _layer="design_template",
  control(T),  /* Execute T (and output) */
  concat(" Sub_design Node: ",T,TITLE),
  repeat,
     menu(10,25,71,23,  ["Introduction",
                          "Do you want to know where you
                          are now ?",
```

```
                         "How reasoned so far ?",
                         "Update      input      and
                          re-evaluate",
                         "Output",
                         "Continue"],
          TITLE,6,CH),
     process(CH,T),
     not(between(CH,1,5)),!.

/*====================================================
     This  is  the  execution  of  Nodes  in  Set-Up  and
   Configuration Phase
====================================================*/

execute_node_template(_,T) :-

   concat("LAYER-NODE: ",T,TITLE),
   repeat,
       menu(10,25,71,23,  ["Introduction",
                  "Do  you  want  to  know  where  you  are
                   now ?",
                  "How Reasoned so far ?",
                  "Continue"],
                  TITLE,4,CH1),
     CH = CH1 + 10,
     process(CH,T),
     not(between(CH,11,13)),!.


execute_node_template(_,_).

/*****************************************************
     PROCESS options for executing Menu of Nodes
*****************************************************/

CLAUSES

/* For Nodes of Design Activity */

   process(0,_).
   process(1,T)  :- view(introduction,4,T).
            /* Explain the nodes in detail. */
   process(2,T)  :- cursor(ROW,COL),show_where(T,ROW,COL).
            /* Explain the present position */
   process(3,_)  :output(reason),view(reason,5,reason),
                   deletefile("reason.rea").
            /* Explain the reasoning processes */
   process(4,T)  :- put_screen(T), control(T).
            /* Execute with new input as desiner wishes */
   process(5,T)  :- view(output,5,T). /* Showing output */
   process(6,_). /* Continue to Next Step */

/* For Nodes of Set-Up and Configuration Phase
   Please refer to the above explanation  */

   process(10,_).
   process(11,T)  :- view(introduction,4,T).
   process(12,T)  :- cursor(ROW,COL),show_where(T,ROW,COL).
```

```
  process(13,_) :-output(reason),view(reason,5,reason),
                  deletefile("reason.rea").
  process(14,_)./* Continue to Next Step */


/********************************************************
  Execution of Design TEMPLATES in Design Activity
********************************************************/

CLAUSES

  control(T) :- concat("Implementing : ",T,MSG),
                put_msg(MSG), init_tmp(T),find(T), fail.
  control(T) :- !,output(T).


/********************************************************
  The Nodes Expressions from Design Analysis
********************************************************/
/*=======================================================
            Paload_Range Template
=======================================================*/

find(payload_range):-
     user_r(range,_r),
     user_r(cargo,_Added_cargo),
     user_r(pax,_p),
     user_r(fineness,_l_d),
     user_r(fuse_wing,_Sf_Sw),
     user_r(size,_L_bar),
     _crews=(_p/30)+3.0,
     _cargo=0.0*_p,
     _payload=(_p+_crews)*(175+40)+_cargo+_Added_cargo,
     _range=_r+500.0,
     assert(datapr(pr,_payload,_range)),
     _At_S=4*_Sf_Sw*(_l_d-1)+3.38,
     _cd0_ref=0.0131*(_At_S/5.0),
     _size=exp((ln(1.0/_L_bar))/6.0),
     _cd0=_cd0_ref*_size,
     assert(datapr(cd0,_cd0,_At_S)),fail.
/*=======================================================
        Landing_performance Template
=======================================================*/
find(landing_performance):-
     user_r(aspect,_aspect),
     user_r(land_d,_land_d),
     user_r(engine_no,_number),
     user_r(range,_range),
     datapr(cd0,_cd0,_),
     dataeng(missed_approach,engine_no,_number,_grad),
     datae(landing,_e),!,
     datacd(landing,_cla,_cdf,_cdg),
     _cd_p=_cd0+_cdf+_cdg,
     _VA=sqrt(_land_d*10.0/3.0),
     _WL_S=((_VA*_VA)/294.1225)*_cla,
     _cdi=_cla*_cla/(3.141592*_aspect*_e),
     _L_D_ratio=_cla/(_cd_p+_cdi),
     _To_WL=(_number/(_number-1))*(1.0/_L_D_ratio+_grad),
```

```
     rrange(_range,_R),
     _Wg_S=_WL_S/_R,
     _To_Wg=_To_WL*_R,
     assert(dataland(landing,_cla,_Wg_S,_To_Wg)),fail.
/*=======================================================
         Takeoff Performance Template
=======================================================*/
find(take_off_performance):-
     user_r(aspect,_aspect),
     user_r(t_o_d,_tod),
     user_r(engine_no,_number),
     datapr(cd0,_cd0,_),
     dataeng(second_climb,engine_no,_number,_grad),
     datae(takeoff,_e),!,
     datacd(takeoff,_cl2,_cdf,_cdg),
     _cd_p=_cd0+_cdf+_cdg,
     _cdi=_cl2*_cl2/(3.141592*_aspect*_e),
     _L_D_ratio=_cl2/(_cd_p+_cdi),
     _To_Wg=(_number/(_number-1))*(1.0/_L_D_ratio+_grad),
     _Wg_S=(_tod/37.6)*(_cl2*1.44)*(_To_Wg),
     assert(datato(takeoff,_cl2,_Wg_S,_To_Wg)),fail.
/*=======================================================
      Cruise Performance Template
=======================================================*/
find(cruise_performance):-
     user_r(aspect,_aspect),
     user_r(mecon,_mcr),
     user_r(bypass,_BPR),
     datae(cruise,_e),
     datapr(cd0,_cd0,_),
     _RR=3.141592*_aspect*_e/_cd0,
     _L_D_max=0.5*sqrt(_RR),
     _cl_max=sqrt((_cd0*3.141592)*_aspect*_e),
     _cl_m1=_cl_max*_mcr*_mcr,
     datawgs(cruise,_Wg_S,_),
     _alt=145427.3*(1-exp((1/5.256)*
          ln(_Wg_S/(_cl_m1*1481.34)))),
     solve(_alt,_mcr,_Tc_To,_,_BPR),
     _To_Wg1=1/(_Tc_To*_L_D_max),
     assert ( datacr (cruise1, _alt,_Wg_S,
              _To_Wg1,_L_D_max)),fail.

find(cruise_performance):-
     user_r(aspect,_aspect),
     user_r(mecon,_mcr),
     user_r(bypass,_BPR),
     datae(cruise,_e),
     datapr(cd0,_cd0,_),!,
     datawgs(cruise,_Wg_S,_K),
     _RR=3.141592*_aspect*_e/_cd0,
     _L_D_max=0.5*sqrt(_RR),
     _cl_max=sqrt((_cd0*3.141592)*_aspect*_e),
     _factor=(2/_K-sqrt(4/(_K*_K)-4))/2,
     _cl_k=_factor*_cl_max,
     _cl_m2=_cl_k*_mcr*_mcr,
     _L_D=_L_D_max*_K,
     _alt=145427.3*(1-exp((1/5.256)*
          ln(_Wg_S/(_cl_m2*1481.34)))),
```

```
    solve(_alt,_mcr,_Tc_To,_,_BPR),
    _To_Wg2=1/(_Tc_To*_L_D),
    assert(datacr(cruise2,_alt,_Wg_S,_To_Wg2,
    _L_D)),fail.
/*============================================
     Size Matching Template
==============================================*/
find(size_matching) :- findpoints,
    intersect1,intersect2,intersect3,intersect4,
    retract(datasiz(_,_,_,_,_)),fail.
find(size_matching):-
    user_r(mecon,_mcr),
    user_r(fuse_wing,_Sf_Sw),
    user_r(aspect,_aspect),
    user_r(fineness,_l_d_r),
    user_r(bypass,_BPR),
    datapr(pr,_Wp,_Range),
    datamat(matching,_Wg_S,_T0_Wg,_Wg_S1,_,_Wg_S2,_),
    datacr(_str,_H1,_Wg_S1,_,_L_D),
    datacr(_str,_H2,_Wg_S2,_,_L_D),
    _H=(_H1*(_Wg_S2-_Wg_S)+_H2*(_Wg_S-_Wg_S1))/(_Wg_S2-
       _Wg_S1),
    speed_of_sound(_H,_Vs),
    _V_knot=_mcr*_Vs,
    solve(_H,_mcr,_,_sfc,_BPR),
    Breguet=(_L_D*_V_knot)/_sfc,
    _Wf_Wg=1-(1/exp(_Range/Breguet)),
    _U_bar=-1.0428*_T0_Wg+0.769,
    Wg=_Wp/(_U_bar-_Wf_Wg),
    change(user_r(aum,Wg)),
    We=Wg*(1-_Wf_Wg)-_Wp,
    Wf=_Wf_Wg*Wg,
    S=Wg/(_Wg_S),
    Span=sqrt(_aspect*S),
    T0=_T0_Wg*Wg,
    Dia=sqrt(4*S*_Sf_Sw/3.141592),
    Length=Dia*_l_d_r,
    assert(datafinal(size_matching,Wg,We,Wf,S,T0,
          Dia,Length,Span,_Wg_S,_T0_Wg,_L_D)),fail.
/*============================================
Airfoil selection template(From Wing Design Program [1]).
==============================================*/
find(airfoil_selection) :-
   wing_section(parameter,_item), /* cls,cl,cd,cm,md */
   section(_S),
     range1(_item, _S,_V),
     assert(data(_item,_S,_V)), fail.

find(airfoil_selection) :-
   score_sections,
   section(SECTION),
     findall(S1,score(_,SECTION,S1),S1s),
     sumList(S1s,TOTAL),
     assert(data(total,SECTION,TOTAL)), fail.

find(airfoil_selection) :- /* Only two <> t/c */
   findall(_S, data(total,_,_S), _Ts),
   maxList(_Ts,_T),
```

```
    data(total,_S1,_T),
    tc(_S1, _TC1),
    section(_S2),
      tc(_S2, _TC2),
      _TC2 <> _TC1,
    assert(data1(best,_S1)),
    assert(data1(second_best,_S2)), !,fail.
/*=======================================================
    3D drag rise template.
=======================================================*/
find(drag_rise_3d) :-
    user_s(fblend,_answer),
    user_r(mecon,_Mecon),
    user_r(mmax,_Mmax),
    wvalue1(fuselage_wing_interaction,_answer,_decr),
    _MDR = _Mecon + _decr,
    assert(data1(fuselage_wing_interaction,_decr)),
    assert(data1(mdr,_MDR)),
    _diff = _Mmax - _MDR,
    _diff <= 0.02,
    assert(high_speed), fail.

find(drag_rise_3d) :-
    wvalue(tc_limit,_limit),
    data1(best,_S1),
    data1(second_best,_S2),
    data1(mdr,_MDR),
    data(md,_S1,_Md1),
    data(md,_S2,_Md2),
    tc(_S1,_TC1),
    tc(_S2,_TC2), !,
    angle(_A),
      d_to_r(_A,_r_A),
      _MDR1 = _Md1 / sqrt(cos(_r_A)),
      _MDR2 = _Md2 / sqrt(cos(_r_A)),
      _TCA  = _TC1*(_MDR  -  _MDR2)/(_MDR1  -  _MDR2)  +
      _TC2*(_MDR - _MDR1)/(_MDR2 - _MDR1),
      assert(data(tc,_A,_TCA)),
      _TCA <= _limit,
      assert(angles(drag_rise_3d_angles,_A)), fail.
*======================================================
     Aeroelastic template.
=======================================================*/
find(aeroelasticity) :-
    wvalue(aspect_ratio_tolerance, _tolerance),
    user_r(altitude,_H),
    user_r(mmax,_Mmax),
    user_r(aspect,_AR0),
    user_s(engine,_answer),
    user_s(active,_answer1),
    wvalue1(engine_position, _answer, _F),
    wvalue1(active_controls,_answer1, _N),
    speed_of_sound(_H,_speed_of_sound),
    _VD = (_Mmax + 0.05) * _speed_of_sound,
    assert(data1(vd,_VD)),
    assert(data1(engine_position,_F)),
    assert(data1(load_factor,_N)),
    _P  = _F * 100000000.0 / (_VD * _VD),
```

```
    _N1 = 850.0 / _N,
    _d  = _AR0 * (100 - _tolerance) / 100, !,
    vangle(aeroelastic,_A),
      aeroelastic(torsion,_P,_A,_AR0,_AR1),
      aeroelastic(bending,_N1,_A,_AR0,_AR2),
      assert(data(torsion,_A,_AR1)),
      assert(data(bending,_A,_AR2)),
      min(_AR1,_AR2,_AR),
      assert(data(aspect_ratio,_A,_AR)),
      _AR >= _d,
      assert(angles(aeroelastic_angles,_A)), fail.

/*=====================================================
       Tip stall template.
=====================================================*/
find(tip_stall) :-
    data1(best,_S1),
    data1(second_best,_S2),
    wing_section(curve,_item),     /* clstall, alfao */
      range1(_item,_S1,_V1),
      range1(_item,_S2,_V2),
      assert(data(_item,_S1,_V1)),
      assert(data(_item,_S2,_V2)), fail.

find(tip_stall) :-  /* CL distributions along the wing */
    wvalue(a0, _A0),
    user_r(mecon,_Mecon),
    user_r(cl,_CL),
    user_r(taper,_TR),
    user_r(wtwist , _Et),
    data1(best,_S),
    data(alfao, _S, _ALFA0),
    data(clstall, _S, _CLs),
    _B1     = 0.5 * _A0 / 57.29578,
    _CLYb1  = _ALFA0 * _B1,
    _CLYb2  = _Et * _B1,
    _CCbar1 = 1.5 / (_TR * _TR + _TR + 1.0),
    _Ybar1  = (10.4  * sqrt(_TR)  - 6.7)  *  (1  -  _Mecon  *
              _Mecon) / 1000,
    _Ybar2 = (4.4 + 5 * _TR) / 1000,
    INC = 1/10, !,
    vangle(tip_stall,_A),
      data(aspect_ratio, _A,_AR),
      d_to_r(_A, _r_A),
      _Ybar = (_Ybar1 + _Ybar2 * tan(_r_A)) * _AR + 0.42,
      for(_N,0,0.9,INC),  /* Loading stations */
        _CLYb  = _CLYb1 - (_CLYb2 * _N),
        _CCbar = (_TR * _TR * _N - _N + 1.0 + _TR) *
                  _CCbar1,
        additional_load_factor(_N, _F),
        _Ya    = (1.28 * sqrt(1.0 - _N * _N)) + (_Ybar -
                 0.425) * _F,
        _CLYa = _CL * _Ya / _CCbar,
        _CLY  = _CLYa + _CLYb,
        assert(load(_A, _N, _CLY)),
        _N >= 0.9,
        LAST = _N + INC,
        assert(load(_A,LAST,0.0)), /* At wing tip */
```

```
        findall(_V,load(_A, _, _V),LCLY),
        maxList(LCLY,MAX),
        MAX < _CLs,      /* Check stall */
        assert(angles(tip_stall_angles,_A)), fail.

/*=======================================================
    Flap template.
=======================================================*/
find(flap) :-
    user_r(cland, _CLmax),
    wvalue(flapcl, _CLf),
    !,
    vangle(flap,_A),
      d_to_r(_A, _r_A),
      _CL = _CLf * cos(_r_A),
      assert(data(maximum_landing_lift_coefficient,
             _A,_CL)),
      _CL >= _CLmax,
      assert(angles(flap_angles,_A)), fail.
/*=======================================================
     Wing weight template.
=======================================================*/
find(wing_weight) :-
    user_r(range,_range),
    user_r(taper,_TR),
    user_r(wloading,_WS),
    user_r(aum,_W),
    data1(vd,_vdKnots),
    data1(load_factor,_N),
    line_eq([500, 9000, 0.034, 0.028, _range],_C1),
    _VD  = _vdKnots * 1852.0 / 3600.0,
    _S   = _W / _WS,
    _P1  = (1 + 2 * _TR) * _S / (3 + 3 * _TR),
    _P2  = sqrt(_VD * _S),
    _PP3 = _N * _WS,
    power(_PP3, 0.3, _P3),
    _PP4 = _P1 * _P2 * _P3,
    power(_PP4, 0.9, _P4),
    _P   = _C1 * _P4, !,
    vangle(wing_weight,_A),
      data(tc, _A, _TCA),
      data(aspect_ratio, _A, _AR),
      d_to_r(_A, _r_A),
      _WP1 = _AR * 100 / (_TCA * 1.4),
      _WP2 = sqrt(_WP1) / cos(_r_A),
      power(_WP2, 0.9, _WP3),
      _WW = _P * _WP3,
      assert(data(wing_weight,_A,_WW)), fail.

find(T):-find1(T).

/* =====================================================
   The followings are the design nodes of Fuselage and
   Engine Selection
/* In case that all the programs can not be run in one
   module, please split the following into new module */
=======================================================*/
```

```
/*project "VEHICLE"
code =1200
include "VGLOBDEF.PRO"*/

PREDICATES
     nondeterm seat_abreast(real,real)
     nondeterm no_of_aisle(real,real)
     nondeterm no_of_access_door(real,real)
     nondeterm range_coeff(real,real)
     nondeterm engine_selection(string,real,real)

CLAUSES

/*=========================================================
     Fuselage's Total Length template.
=========================================================*/

find1(total_length):-
    user_r(pax,_pax),
    user_r(gap,_gap),
    user_r(thickness,_thickness),
    user_r(pitch,_pitch),
    user_r(arm_rest_width,_arm_rest_width),
    user_r(seat_width,_seat_width),
    user_r(aisle_width,_aisle_width),
    user_r(nose_ld,_nose_ld),
    user_r(tail_ld,_tail_ld),!,
    seat_abreast(_pax,_SA),
    no_of_aisle(_pax,_no_of_aisle),
    no_of_access_door(_pax,_no_of_access),
    _diameter=(_seat_width/12)*_SA+
         (_arm_rest_width/12)*(_SA+_no_of_aisle+1)+

         (_aisle_width/12)*_no_of_aisle+(_gap/12+_thickne
           ss/12)*2,
    _long=(_pax/_SA)*(_pitch/12)+_no_of_access*(10/12),
    _length=_long+_diameter*(_nose_ld+_tail_ld),
    _long1=_long+(1.5+2.5)*_diameter,
    _long2=_long+(2.0+3.0)*_diameter,
    between(_length,_long1,_long2),
    _LD=_length/_diameter,
    between(_LD,6.8,15.0),
    change(user_r(fineness,_LD)),
    assert(data_all(fuselage,diameter,_diameter)),
    assert(data_all(fuselage,length,_length)), fail.
/*=========================================================
     Fuselage's Weight template.
=========================================================*/
find1(fuselage_weight):-
    data_all(_,diameter,_D),
    data_all(_,length,_L),
    user_r(altitude,_H),
    user_r(range,_R),
    user_r(mmax,_MCR),
    speed_of_sound(_H,_Vs),
    range_coeff(_R,_C2),
    _VD=(_MCR+0.05)*_Vs,
    assertz(data_all(fuselage,design_diving_speed,_VD)),
```

```
    _VD1=sqrt(_VD),
    _MA=2*(0.3048*_L)*(0.3048*_D)*_VD1,
    power(_MA,1.5,_MAS),
    _MASS=_C2*_MAS*2.202643172,
    composite_factor(fuselage,_composite),
    _MASS1=_MASS*(1-_composite),
    assertz(data_weight(fuselage,metal,_MASS)),
    assertz(data_weight(fuselage,composite,_MASS1)),fail.
/*======================================================
     Engine Selection template.
======================================================*/
find1(engine_selection):-
    user_r("engine_no",_total_engines),
    user_r("bypass",_BPR),
    datafinal(size_matching,_,_,_,_,_T0,_,_,_,_,_),
    _thrust_per_engine=_T0/_total_engines,
    findall(ENGINE,engine_selection(ENGINE,_BPR,
            _thrust_per_engine),ENGINES),
    query(single,engine_selection,ENGINES,ANS),
    ANS=[_selected_engine],
    engine_list(_,_selected_engine,_thrust_of_engine,
                _bypass,_weight),
    selecte3(engine,position,_position),
    wvalue1(engine,_position,_factor),
    _engine_weight=_weight*_total_engines*_factor,
    _engine_weight1=_engine_weight*(1-0.03),
    assertz(data_all(_selected_engine,engine,_thrust_per_
            engine)),
    assertz(data_weight(engine,metal,_engine_weight)),
    assertz(data_weight(engine,composite,
            _engine_weight1)),
    assertz(datafinal(engine,_T0,_total_engines,
            _thrust_per_engine,_thrust_per_engine,
            _thrust_of_engine,_BPR,_bypass,_weight,
            _engine_weight,_engine_weight1,
            _engine_weight1)),fail.

find1(_):-!.
/*======================================================
     Engine Selection From the Data.
======================================================*/
engine_selection(ENGINE,_BPR,_thrust_per_engine):-
      engine_list(engine,ENGINE,_thrust,_bypass,_),
      _BPR<=5.25,_bypass<=5.25,
      _ratio=_BPR/_bypass,
      0.7<_ratio,_ratio<1.3,
      _thrust1=_thrust*0.95,
      _KK=((_thrust_per_engine-_thrust)/_thrust1)*100,
      _KKK=abs(_KK), 0<_KKK,_KKK<10.

engine_selection(ENGINE,_BPR,_thrust_per_engine):-
      engine_list(engine,ENGINE,_thrust,_bypass,_),
      _BPR>5.25,_bypass>5.25,
      _ratio=_BPR/_bypass,
      0.7<_ratio,_ratio<1.3,
      _thrust1=_thrust*0.95,
      _KK=((_thrust_per_engine-_thrust)/_thrust1)*100,
      _KKK=abs(_KK),
```

```
       0<_KKK,_KKK<10.

range_coeff(_R,K):-
             _R<2500,/*selecte3(engine,_,
              under_wing_mounted),*/K=0.027.
range_coeff(_R,K):-
             _R>=2500,/*selecte3(engine,_,
              under_wing_mounted),*/K=0.024.
/*range_coeff(_R,K):-
             _R<2500,selecte3(engine,_,
              rear_fuselage_mounted),K=0.0297.
range_coeff(_R,K):-
             _R>=2500,selecte3(engine,_,
              rear_fuselage_mounted),K=0.0264.*/
/*=======================================================
     Seat Abreast Selection.
=======================================================*/
seat_abreast(_pax,_SA):-_pax<=66,_SA=4.
seat_abreast(_pax,_SA):-66<_pax,_pax<=130,_SA=5.
seat_abreast(_pax,_SA):-130<_pax,_pax<=260,_SA=6.
seat_abreast(_pax,_SA):-260<_pax,_pax<=420,_SA=8.
seat_abreast(_pax,_SA):-420<_pax,_pax<500,_SA=9.
seat_abreast(_pax,_SA):-500<=_pax,_SA=10.
/*=======================================================
     Number of Aisle Selection.
=======================================================*/
no_of_aisle(_pax,_no_of_aisle):-_pax<=260,
                                _no_of_aisle=1.
no_of_aisle(_pax,_no_of_aisle):-_pax>260, _no_of_aisle=2.

no_of_access_door(_pax,_no_of_door):-
                                _pax<80,_no_of_door=1.
no_of_access_door(_pax,_no_of_door):-
                      80<=_pax,_pax<200,_no_of_door=2.
no_of_access_door(_pax,_no_of_door):-
                      200<_pax,_no_of_door=3.
/*****************************************************
 Finding the intersection points in parametric study
*****************************************************/
findpoints:- findall(P,gpoint(landing,[_],P),Ps),
             findall(Q,gpoint(takeoff,[_],Q),Qs),
             findall(R,gpoint(cruise1,[_],R),Rs),
             findall(S,gpoint(cruise2,[_],S),Ss),
             select(landing,Ps),select(takeoff,Qs),
             select(cruise1,Rs),select(cruise2,Ss).

intersect1:- datasiz(landing,P1,P2,P3,P4),
             datasiz(cruise1,P5,P6,P7,P8),
             intersect(P1,P2,P3,P4,P5,P6,P7,P8), fail.
intersect1:-!.

intersect2:- datasiz(landing,Q1,Q2,Q3,Q4),
             datasiz(cruise2,Q5,Q6,Q7,Q8),
             intersect(Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8), fail.
intersect2:-!.

intersect3:- datasiz(takeoff,R1,R2,R3,R4),
             datasiz(cruise1,R5,R6,R7,R8),
```

```
             intersect(R1,R2,R3,R4,R5,R6,R7,R8), fail.
intersect3:-!.

intersect4:- datasiz(takeoff,S1,S2,S3,S4),
             datasiz(cruise2,S5,S6,S7,S8),
             intersect(S1,S2,S3,S4,S5,S6,S7,S8), fail.
intersect4:-!.

intersect(_x1,_y1,_x2,_y2,_x3,_y3,_x4,_y4):-
              _xi=((_y3-_y1)+(_y2-_y1)/(_x2-_x1)*_x1-(_y4-
         _y3)/(_x4-_x3)*_x3)
            /((_y2-_y1)/(_x2-_x1)-(_y4-_y3)/(_x4-_x3)),
         _yi=(((_y2-_y1)/(_x2-_x1))*(_xi-_x1)+_y1),
          between(_xi,_x1,_x2),between(_yi,_y1,_y2),
          between(_xi,_x3,_x4),between(_yi,_y4,_y3),
          assert(datamat(matching,_xi,_yi,_x3,_y3,_x4,
                 _y4)).

intersect(_,_,_,_,_,_,_,_):-!.

select(_str,[p(X,Y)|_T]):- not(_T=[]),
           select1(_T,p(X1,Y1)),
           assert(datasiz(_str,X,Y,X1,Y1)),
           select(_str,_T).
select(_,[_]):-!.
select1([p(A,B)|_],p(A,B)):-!.

rrange(_range,_R):- _range<2000,_R=0.91.
rrange(_range,_R):- _range>=2000,_range<3000,_R=0.82.
rrange(_range,_R):- _range>=3000,_R=0.73.

gpoint(landing,[_cla],p(X,Y)):-
       dataland(landing,_cla,X,Y).
gpoint(takeoff,[_cl2],p(X,Y)):-datato(takeoff,_cl2,X,Y).
gpoint(cruise1,[_H1],p(X,Y)):-datacr(cruise1,_H1,X,Y,_).
gpoint(cruise2,[_H2],p(X,Y)):-datacr(cruise2,_H2,X,Y,_).

/*------------------------------------------------------
 Solve  Thrust  Loading,  Specific  Fuel  Consumption  vs
 height. This was obtained from reference [7]
------------------------------------------------------*/
solve(_alt,_mcr,_Tc_To,_sfc,_BPR):-
     _BPR<=5.25,
     _alt<35000,
     _Tc_To=(1/5000)*((_alt-30000)*
            (0.0166667*_mcr+0.20333332)
             +(35000-_alt)*(-0.0166667*_mcr+0.268333)),
     _sfc=(1/10000)*((_alt-20000)*(0.2925*_mcr+0.4185)
         +(30000-_alt)*(0.40625*_mcr+0.345625)).

solve(_alt,_mcr,_Tc_To,_sfc,_BPR):-
     _BPR<=5.25,
     _alt>=35000,
     _Tc_To=(1/5000)*((_alt-35000)*
            (0.022222*_mcr+0.1561111)
           +(40000-_alt)*(0.0166667*_mcr+0.20333332)),
     _sfc=(1/10000)*((_alt-20000)*(0.2925*_mcr+0.4185)
         +(30000-_alt)*(0.40625*_mcr+0.345625)).
```

```
solve(_alt,_mcr,_Tc_To,_sfc,_BPR):-
    _BPR>5.25,
    _alt<35000,
    _Tc_To=(1/5000)*((_alt-30000)*
          (0.08333*_mcr+0.158333)
         +(35000-_alt)*(-0.016667*_mcr+0.26333)),
    _sfc=(1/10000)*((_alt-25000)*(0.3916667*_mcr+0.3125)
        +(35000-_alt)*(0.45*_mcr+0.285)).

solve(_alt,_mcr,_Tc_To,_sfc,_BPR):-
    _BPR>5.25,
    _alt>=35000,
    _Tc_To=(1/5000)*((_alt-35000)*(0.05*_mcr+0.14)
         +(40000-_alt)*(0.08333*_mcr+0.158333)),
    _sfc=(1/10000)*((_alt-25000)*(0.3916667*_mcr+0.3125)
        +(35000-_alt)*(0.45*_mcr+0.285)).

/***********************************************
                RANGE
***********************************************/
range1(cls,_S,_V) :- !, dataw1(cls, _S, _, _V).
range1(cl,_S,_V)  :- !, dataw1(cl, _S, _, _V).

range1(cd,_S,_Y) :- !,
    user_r(mecon,_X),
    user_r(cl,_K),
    findall(_XK, dataw(cd, _S, _XK, _), _Ks),
    test_k(_Ks, _K, _K1, _K2),
    dataw(cd, _S, _K1, _C1s),
    dataw(cd, _S, _K2, _C2s),
    polyn(_X, _C1s, _Y1),
    polyn(_X, _C2s, _Y2),
    line_eq([_K1, _K2, _Y1, _Y2, _K], _Y).

range1(cm,_S,_Y) :- !,
    user_r(mecon,_K),
    user_r(cl,_X),
    findall(_XK, dataw(cm, _S, _XK, _), _Ks),
    test_k(_Ks, _K, _K1, _K2),
    dataw(cm, _S, _K1, _C1s),
    dataw(cm, _S, _K2, _C2s),
    polyn(_X, _C1s, _Y1),
    polyn(_X, _C2s, _Y2),
    line_eq([_K1, _K2, _Y1, _Y2, _K], _Y).

range1(md,_S,_Y) :-!,
    user_r(mecon,_X),
    dataw(md, _S, _, _Cs),
    polyn(_X, _Cs,_Y).

range1(_I,_S,_Y) :-
    wing_section(curve,_C),  /* clstall, alfao */
    _C = _I,
    !,
    data(md,_S,_Md),
    findall(_XM, dataw1(_I, _S, _XM, _), _LM),
    test_k(_LM, _Md, _M1, _M2),
```

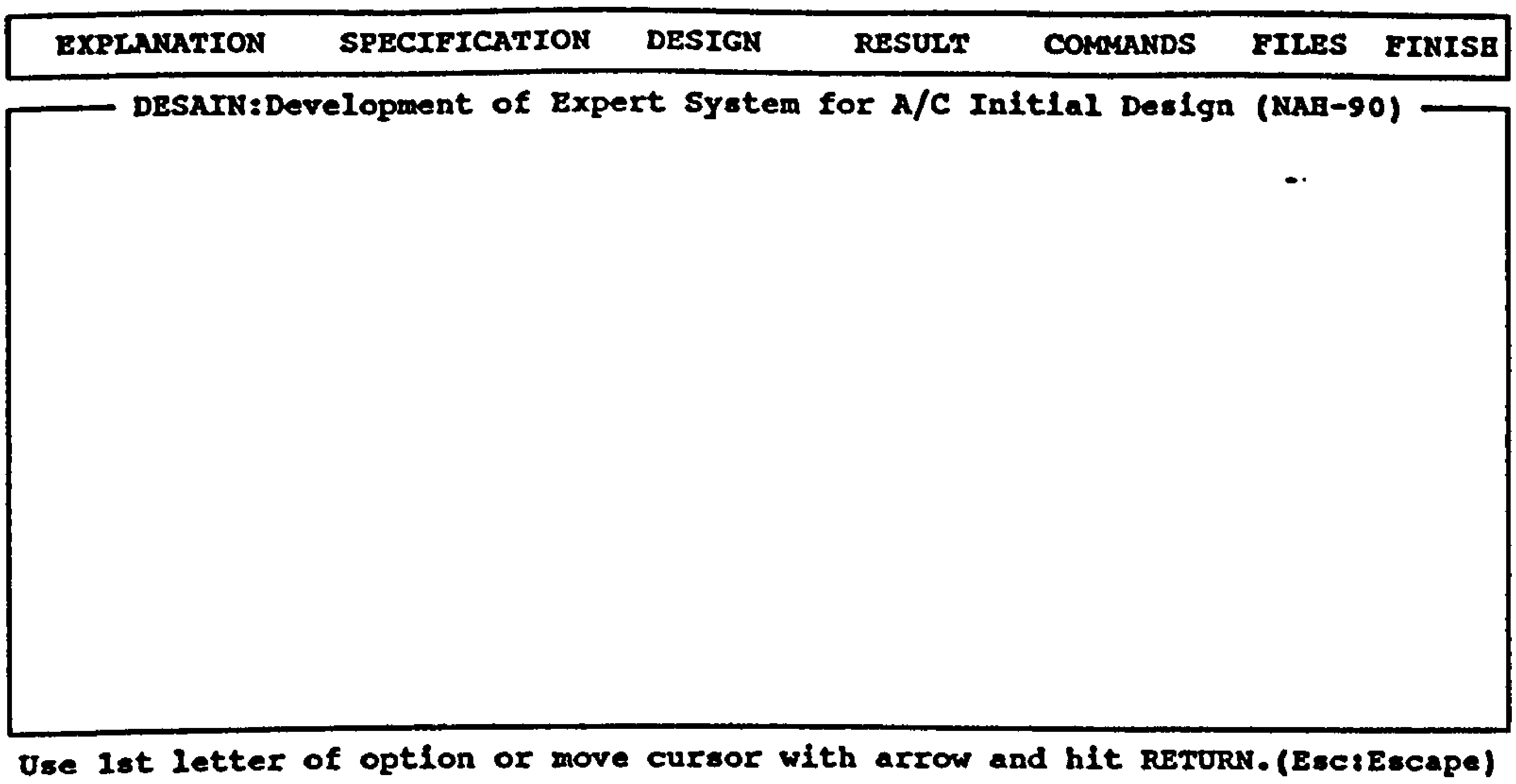
```
    dataw1(_I,_S,_M1,_V1),
    dataw1(_I,_S,_M2,_V2),
    _Y =  _V1*(_Md  -  _M2)/(_M1  -  _M2)  +  _V2*(_Md  -
         _M1)/(_M2 - _M1).

/*****************************************************
*****************************************************
         General ALGORITHMs.

       1. Load Factor
       2. Radian to Degree
       3. Polynomial Equations
       4. Speed of Sound
       5. Aeroelastic Check
       6. Score each Airfoil and Select the best one
       7. Valid Sweep Angle Check
*****************************************************
****************************************************/

additional_load_factor(_N, _F) :- _N < 0.7, !, _F = 14.13
                                  * _N - 6.35.
additional_load_factor(_N, _F) :- !, _F = 4.25 - (_N -
                                  0.815)*(_N-0.815)*53.8 .

d_to_r(_degrees,_radians) :-   _radians =  (3.1415926 *
_degrees) / 180 .

line_eq([_X, _X, _Y, _Y, _X], _Y) :- !.
line_eq([_X1, _X2, _Y1, _Y2, _X], _Y) :- !,
  _Y =  _Y1*(_X - _X2)/(_X1 - _X2) + _Y2*(_X - _X1)/(_X2
       - _X1).

speed_of_sound(_H,_ao) :-  _H <= 36000, !,
  _T = 288.2 - 0.00198 * _H,
  _ao = 1.9438 * sqrt(401.8 * _T).
speed_of_sound(_H, _ao) :- !,_H>36000,
  _ao = 1.9438 * sqrt(401.8 * 216.7).

aeroelastic(torsion, _P, _A,_AR0,_AR) :- !,
    data(tc,_A,_TCA),
    d_to_r(_A,_r_A),
    _TCR = 1.4 * _TCA,
    _AR11 = _TCR * _TCR * _P / cos(_r_A) * 10000,
    power(_AR11, 0.6667, _AR1),
    min(_AR0, _AR1, _AR).

aeroelastic(bending, _N1, _A,_AR0,_AR) :- !,
    data(tc,_A,_TCA),
    d_to_r(_A,_r_A),
    _TCR = 1.4 * _TCA,
    _AR11 = _N1 * _TCR * cos(_r_A) / 100.0,
    power(_AR11, 0.6667, _AR1),
    min(_AR0, _AR1, _AR).

  score_sections :-
      wvalue1(cd, significant_difference, _sig),
      wvalue1(cd, importance, _imp),
      findall(V,data(cd,_,V),Vs),
```

```
    maxList(Vs,WORST),
    section(_S),
      data(cd, _S, _V),
      _diff = WORST - _V,
      _score =  (_imp / _sig) * _diff,
      assert(score(cd, _S, _score)), fail.

score_sections :-
  wing_section(parameter,_item), /* cls, cl, cm, md */
  not( _item = cd ),
    wvaluel(_item, significant_difference, _sig),
    wvaluel(_item, importance, _imp),
    findall(V,data(_item,_,V),Vs),
    minList(Vs,WORST),
    section(_S),
    data(_item, _S, _V),
    _diff = _V - WORST,
    _score =  (_imp / _sig) * _diff,
    assert(score(_item, _S, _score)), fail.

score_sections :- !.

vangle(aeroelastic,_A):-angles(drag_rise_3d_angles,_A).
vangle(tip_stall,_A) :- angles(aeroelastic_angles,_A).
vangle(wing_weight,_A):-angles(aeroelastic_angles,_A).
vangle(flap,_A) :- angles(drag_rise_3d_angles,_A).
```

APPENDIX VI

TRIAL IMPLEMENTATION

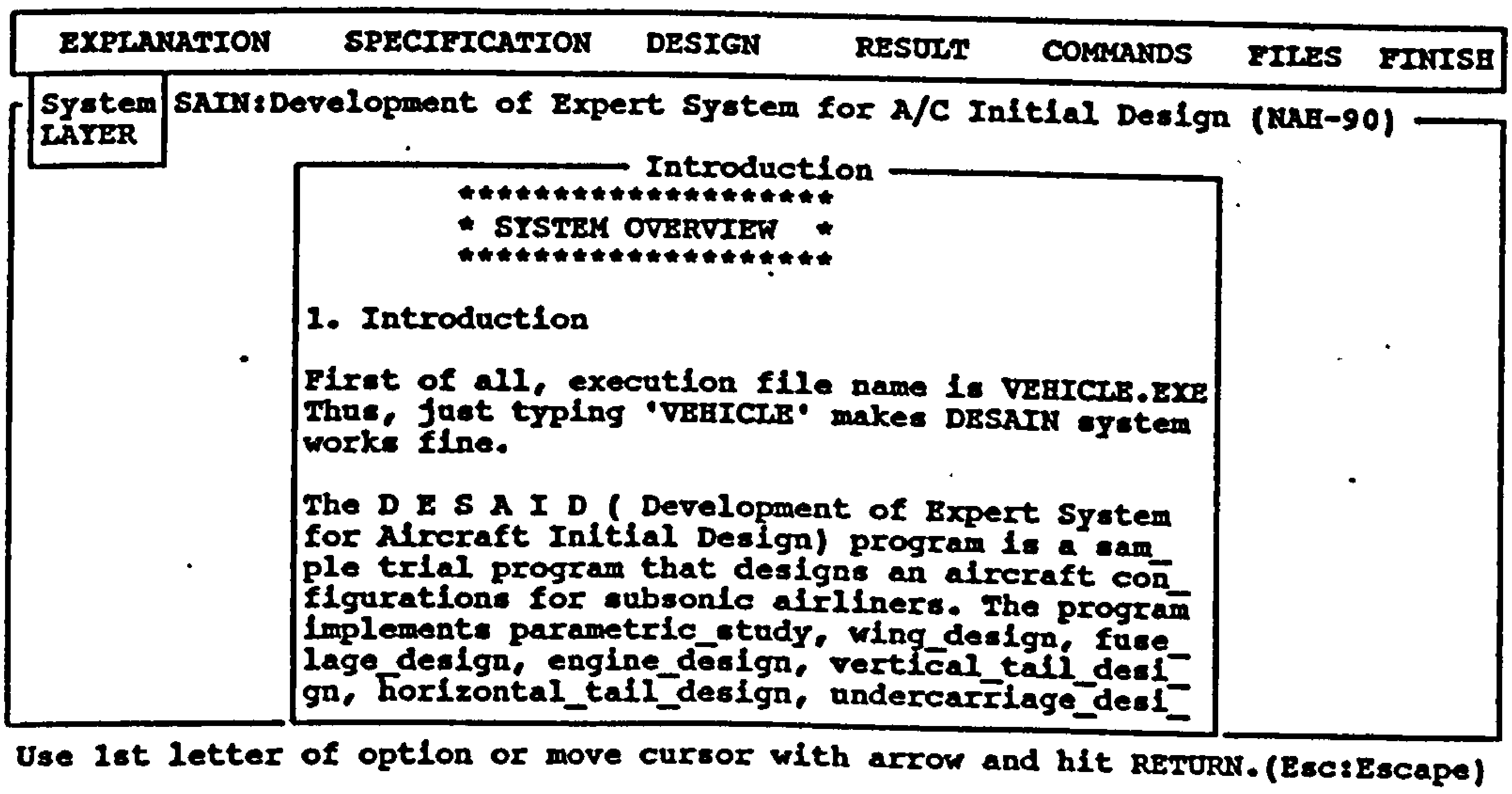
1. If the key "VEHICLE" is pressed, the system shows five pull-down menus, that is, EXPLANATION, SPECIFICATION, DESIGN, RESULTS, FILES, and FINISH, as shown in the following screen. Selecting an option is to use 1st letter of option or to move cursor with arrow and hit return. This was written in the status line positioned at the bottom of the screen.



```
 EXPLANATION    SPECIFICATION   DESIGN     RESULT    COMMANDS    FILES  FINISH
 ---- DESAIN:Development of Expert System for A/C Initial Design (NAH-90) ----
```

Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)

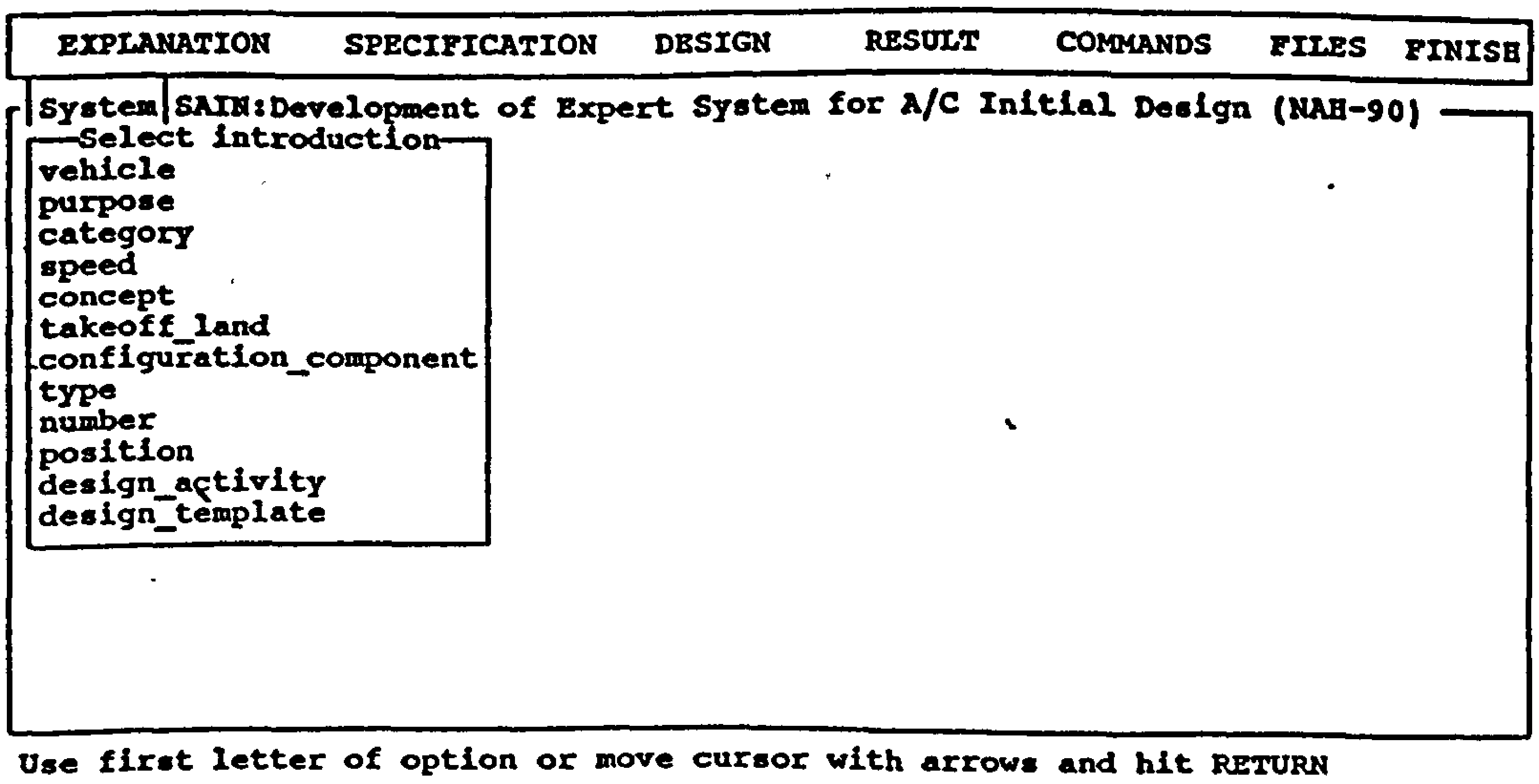
2. The Explanation has 2 menus, System and Layer. If the System is selected, a system file (SYSTEM.TXT) is shown.



```
 EXPLANATION    SPECIFICATION   DESIGN     RESULT    COMMANDS    FILES  FINISH
 System SAIN:Development of Expert System for A/C Initial Design (NAH-90) ----
 LAYER
             ------------ Introduction ------------
                     *********************
                     * SYSTEM OVERVIEW   *
                     *********************

             1. Introduction

             First of all, execution file name is VEHICLE.EXE
             Thus, just typing 'VEHICLE' makes DESAIN system
             works fine.

             The D E S A I D ( Development of Expert System
             for Aircraft Initial Design) program is a sam_
             ple trial program that designs an aircraft con_
             figurations for subsonic airliners. The program
             implements parametric_study, wing_design, fuse_
             lage_design, engine_design, vertical_tail_desi_
             gn, horizontal_tail_design, undercarriage_desi_
```

Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)

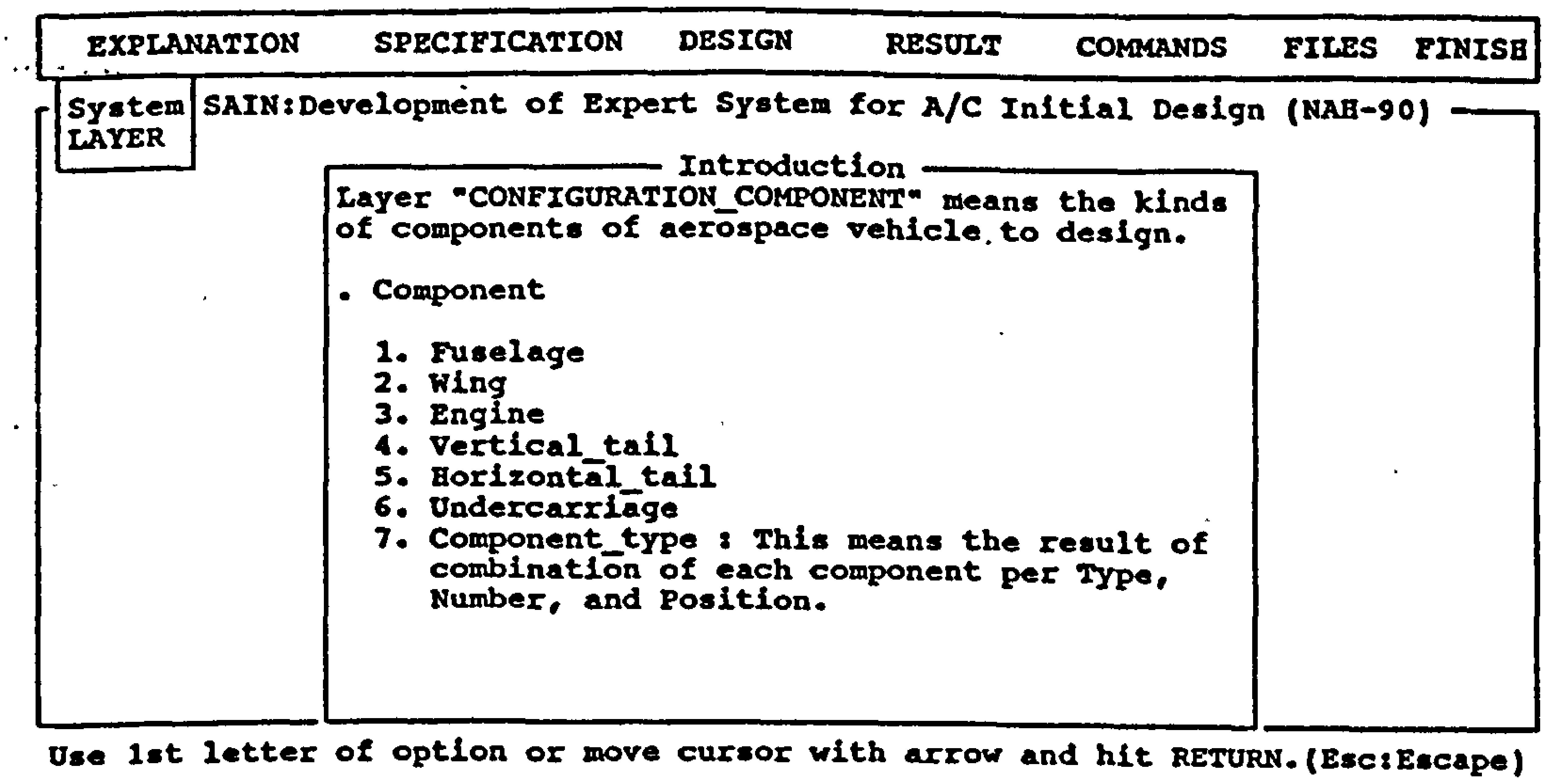
If the Layer is selected, all the layers of DESAID appears on the screen.



SAIN:Development of Expert System for A/C Initial Design (NAH-90)

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

If the layer 'configuration_component' is selected, its associated file which explains the layer appears on the screen.



SAIN:Development of Expert System for A/C Initial Design (NAH-90)

Introduction

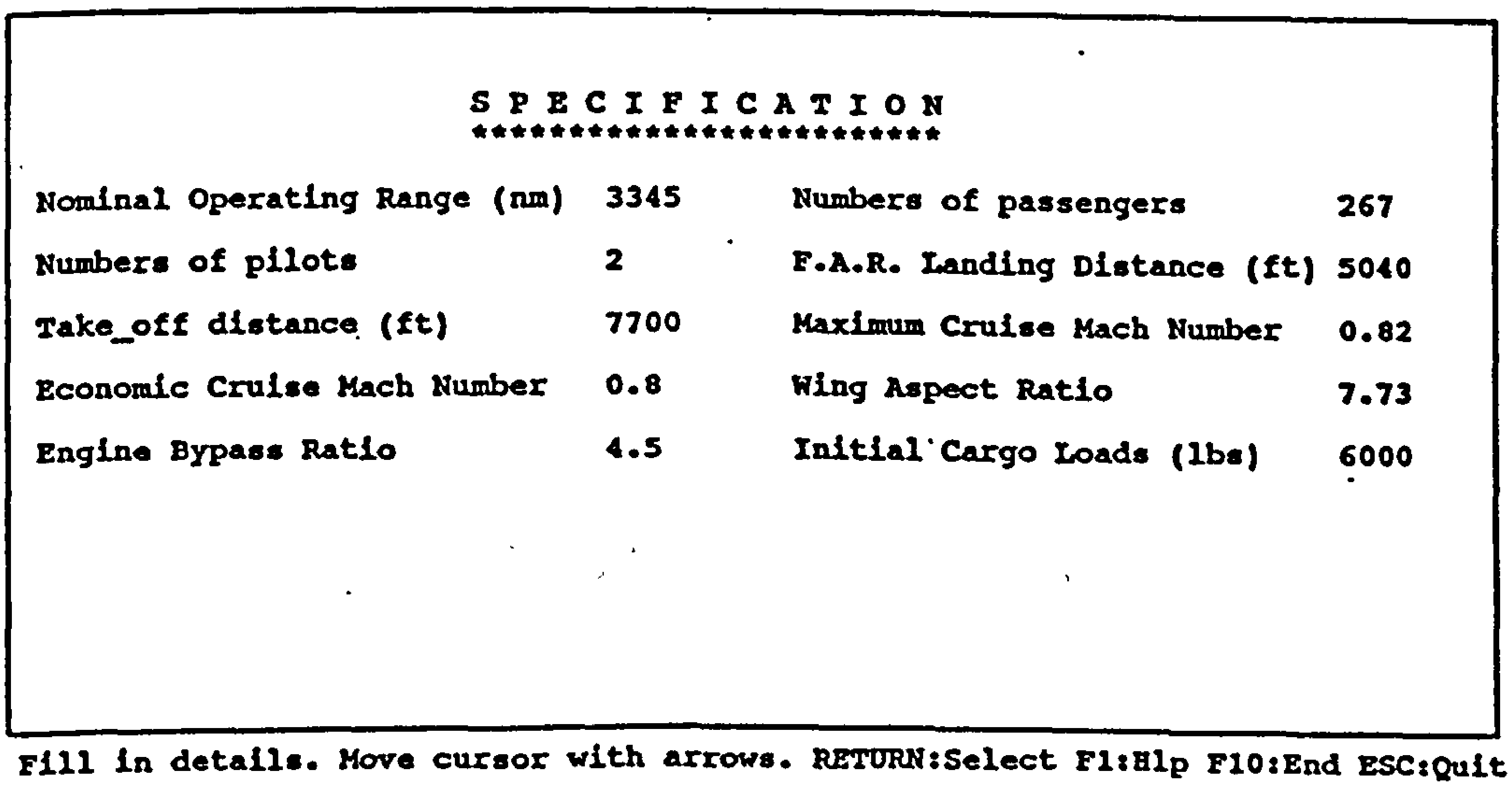
Layer "CONFIGURATION_COMPONENT" means the kinds of components of aerospace vehicle to design.

- Component

  1. Fuselage
  2. Wing
  3. Engine
  4. Vertical_tail
  5. Horizontal_tail
  6. Undercarriage
  7. Component_type : This means the result of combination of each component per Type, Number, and Position.

Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)

3. The Specification shows all the Aircraft requirements and the proper values can be input.

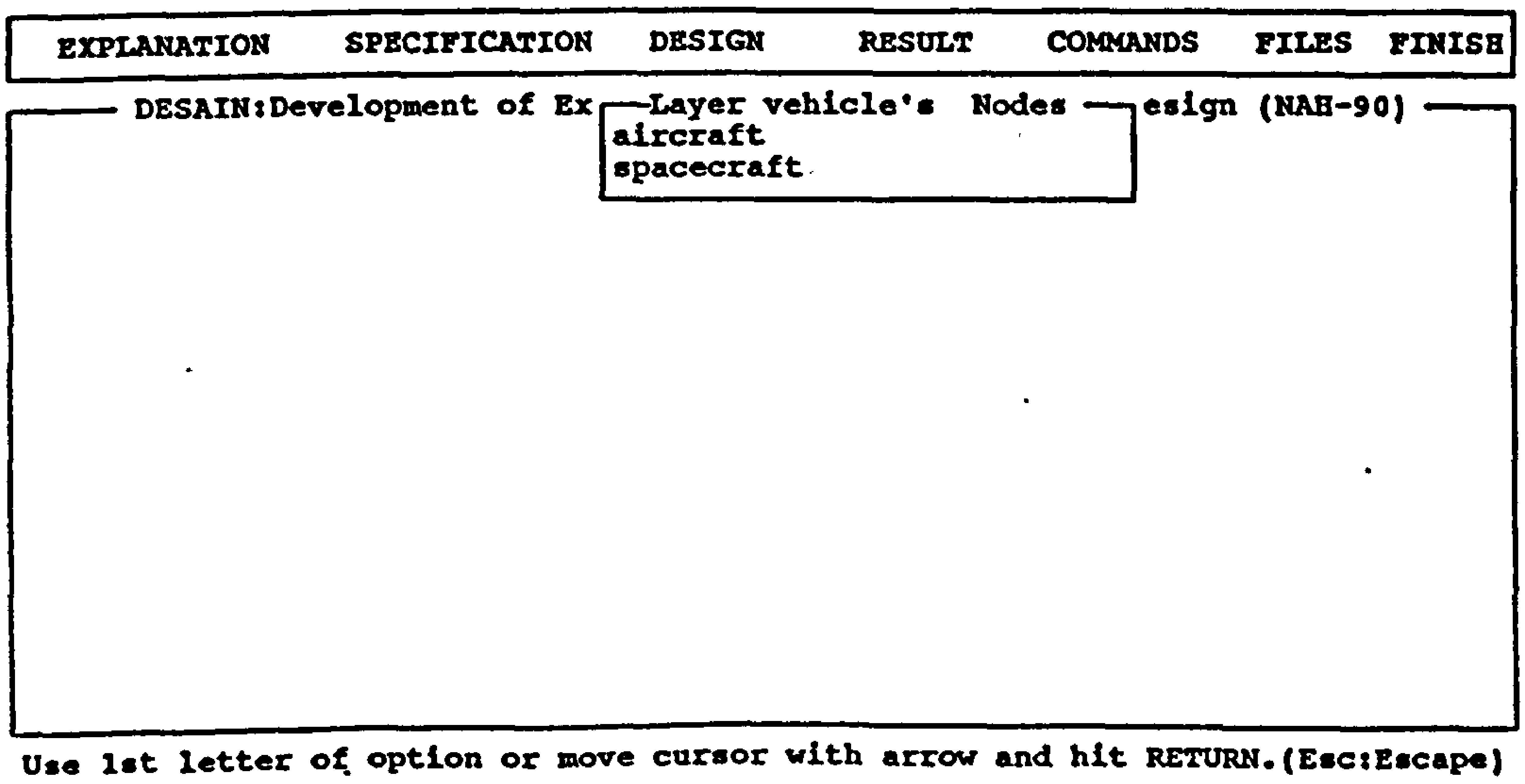


SPECIFICATION

Nominal Operating Range (nm)	3345	Numbers of passengers	267
Numbers of pilots	2	F.A.R. Landing Distance (ft)	5040
Take_off distance (ft)	7700	Maximum Cruise Mach Number	0.82
Economic Cruise Mach Number	0.8	Wing Aspect Ratio	7.73
Engine Bypass Ratio	4.5	Initial Cargo Loads (lbs)	6000

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

4. If the specification input is completed, the cursor will move to the "DESIGN". If the user hits "Return" key, the nodes of the first layer "vehicle" (i.e., aircraft, spacecraft) will appear for his selection.

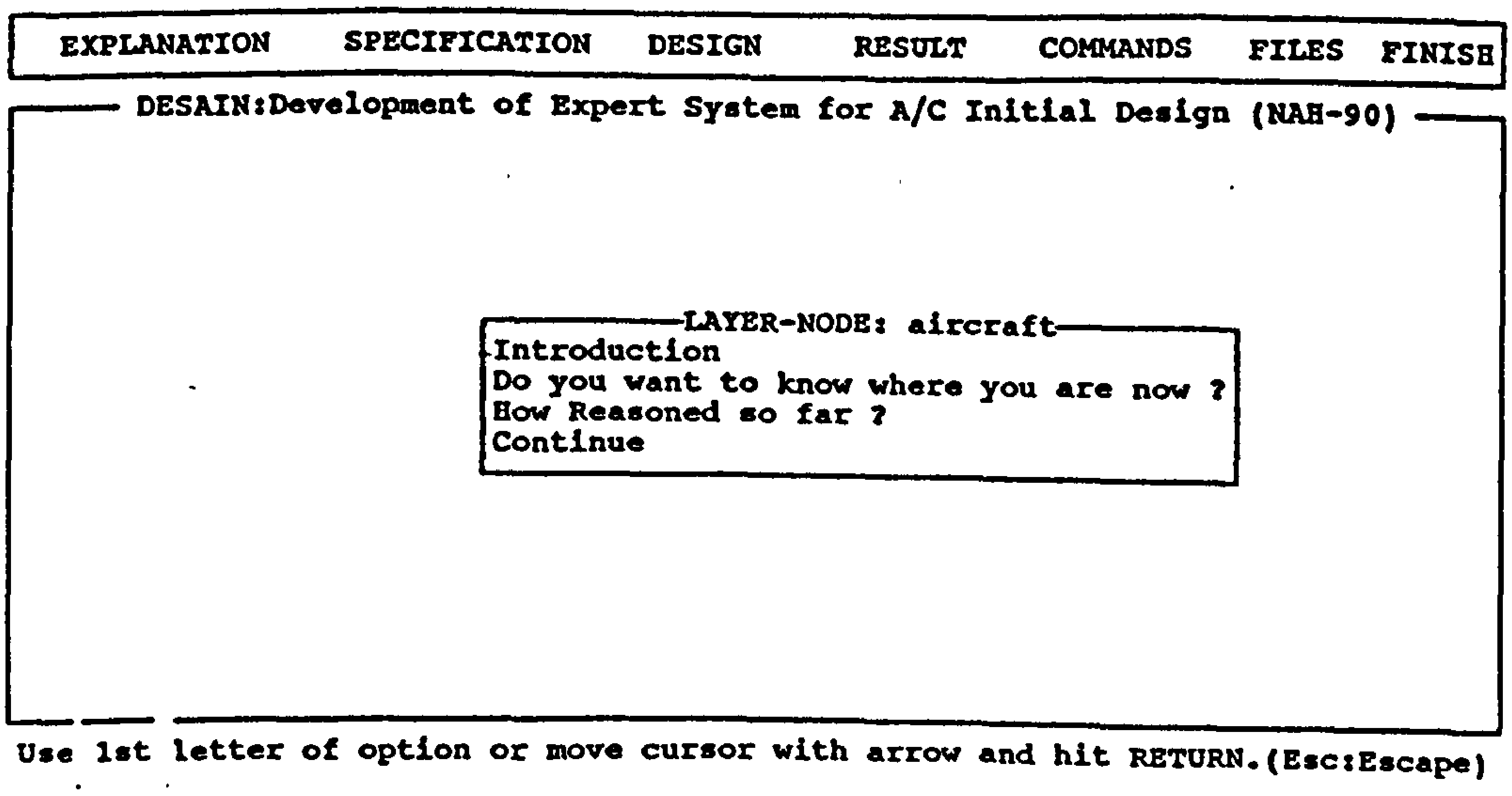


DESAIN:Development of Ex — Layer vehicle's Nodes — esign (NAH-90)

aircraft
spacecraft

Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)

5. If the designer selects the node "aircraft", the Node's menu of 4 items will appear in the screen.



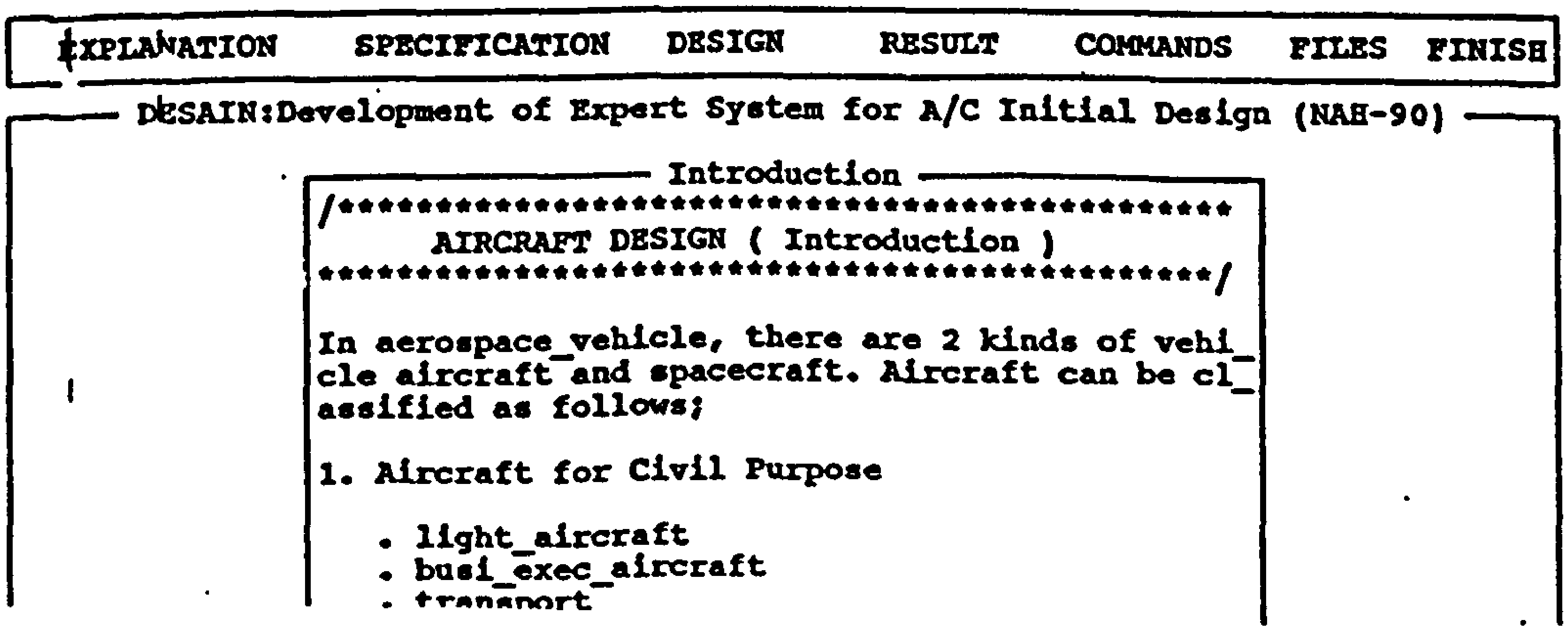
DESAIN:Development of Expert System for A/C Initial Design (NAH-90)

LAYER-NODE: aircraft

Introduction
Do you want to know where you are now ?
How Reasoned so far ?
Continue

Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)

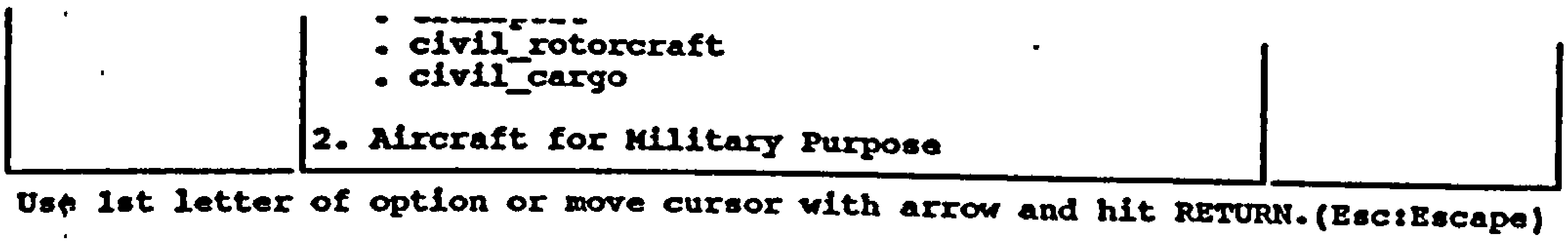
6. If you select "Introduction" from the above, the file (aircraft.txt) appears in the screen.



DESAIN:Development of Expert System for A/C Initial Design (NAH-90)

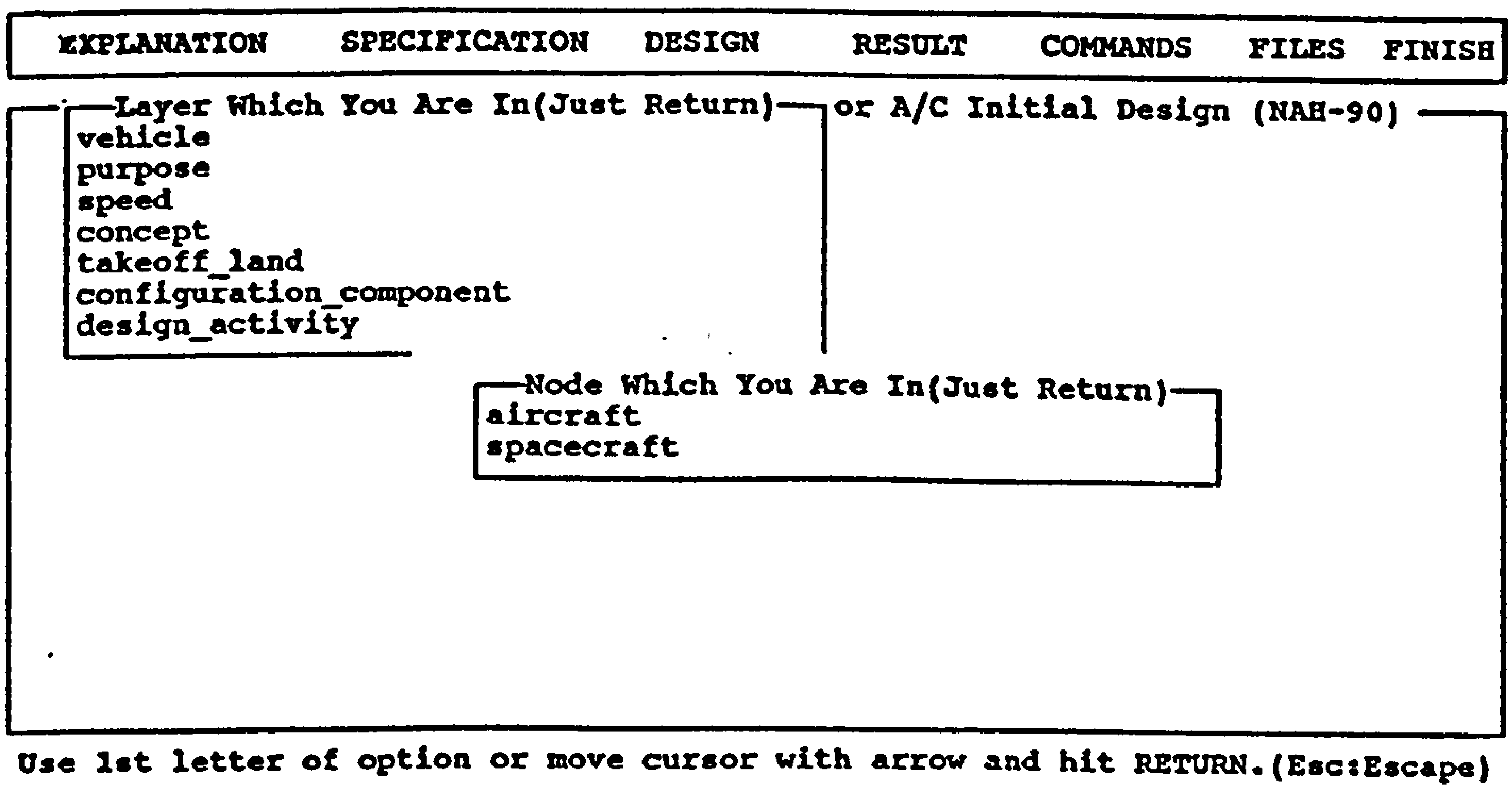
Introduction

```
/**********************************************
       AIRCRAFT DESIGN ( Introduction )
**********************************************/

In aerospace_vehicle, there are 2 kinds of vehi_
cle aircraft and spacecraft. Aircraft can be cl_
assified as follows;

1. Aircraft for Civil Purpose

   . light_aircraft
   . busi_exec_aircraft
   . transport
```



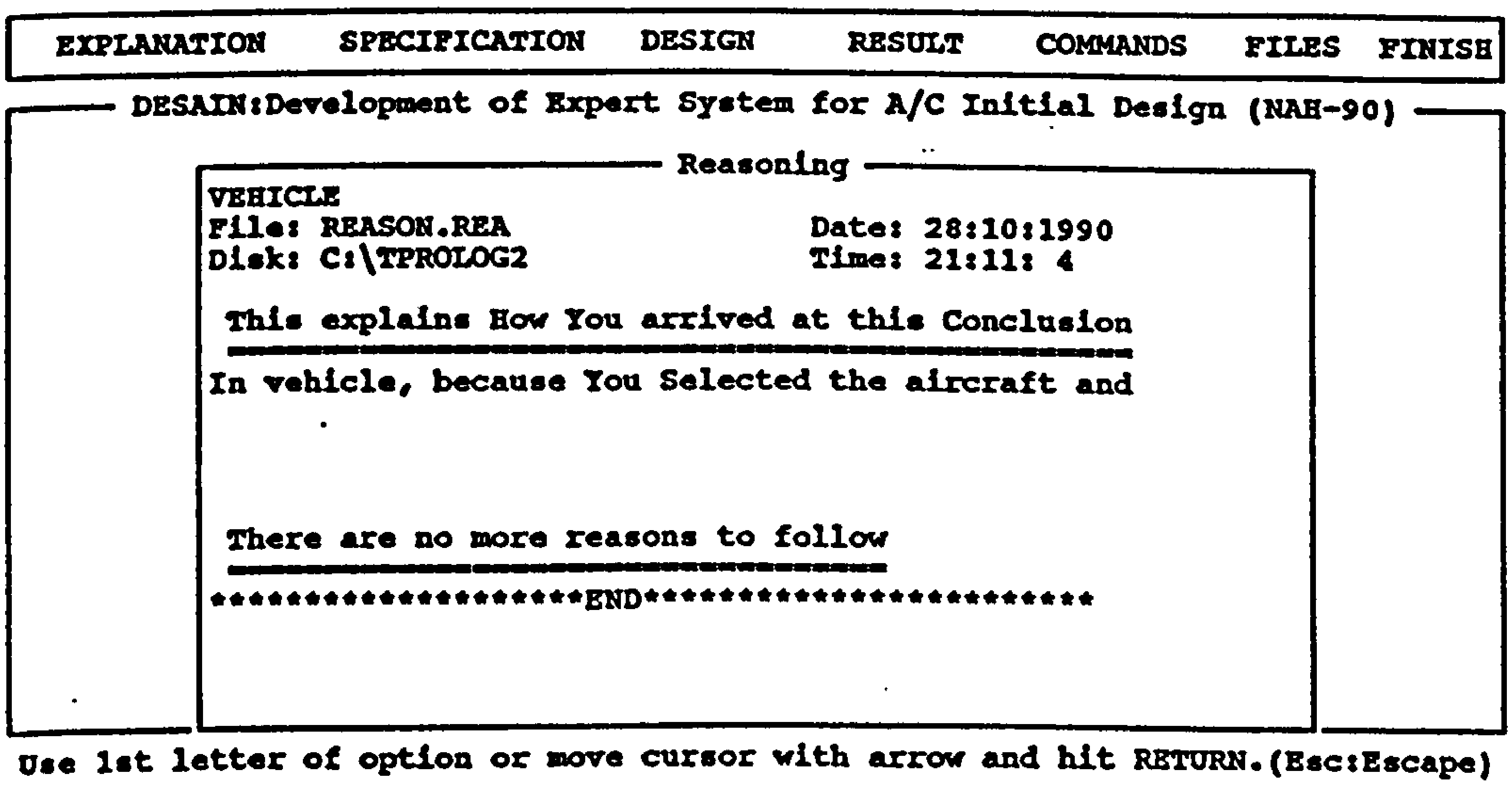
```
  • civil_rotorcraft
  • civil_cargo

2. Aircraft for Military Purpose
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```

7. If you select "Do you want to know where you are now ?", the current layer and node are shown where he is now.



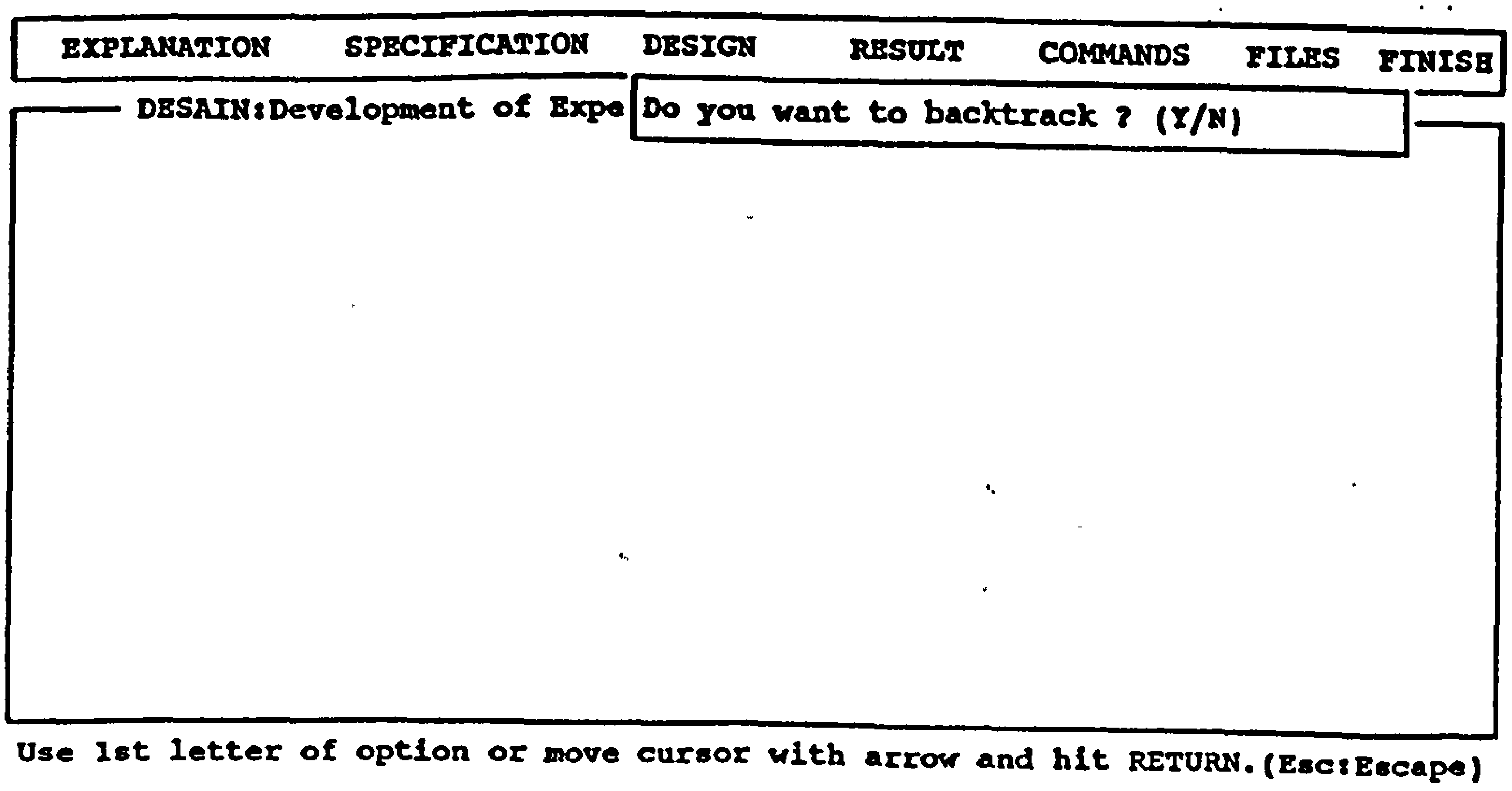
```
Layer Which You Are In(Just Return)     or A/C Initial Design (NAH-90)
vehicle
purpose
speed
concept
takeoff_land
configuration_component
design_activity

               Node Which You Are In(Just Return)
               aircraft
               spacecraft

Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```

8. If you select "How reasoned so far ?", the reasoning file (aircraft.rea) which explains to you the reasoning process will be shown.



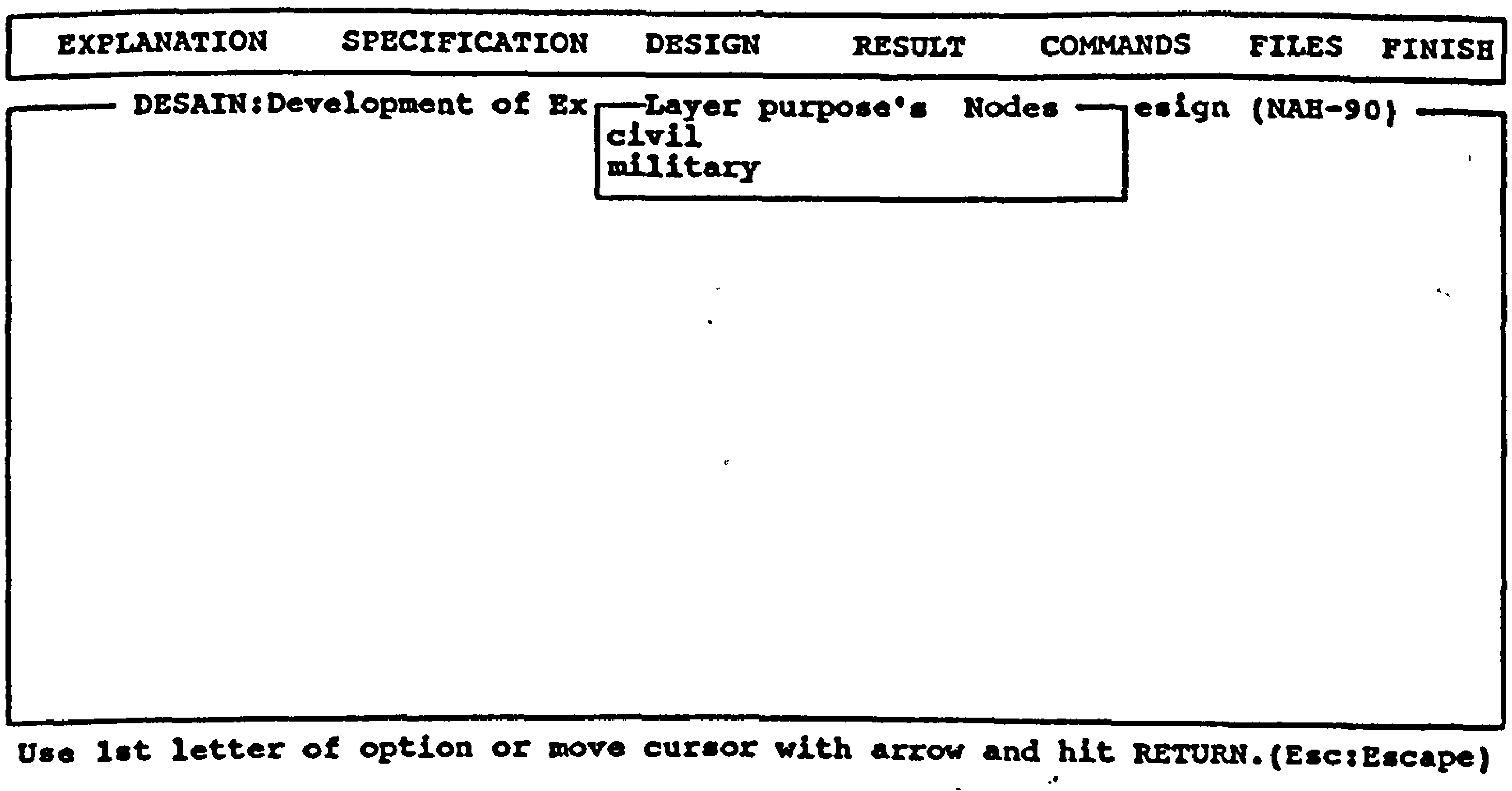
```
DESAIN:Development of Expert System for A/C Initial Design (NAH-90)
                         Reasoning
VEHICLE
File: REASON.REA                   Date: 28:10:1990
Disk: C:\TPROLOG2                  Time: 21:11: 4

 This explains How You arrived at this Conclusion
 ================================================
In vehicle, because You Selected the aircraft and


 There are no more reasons to follow
 ===================================
*******************END***********************

Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```

9. If you select "continue" which proceeds to a next step, the system asks whether you want to backtrack to the previous nodes. If you type "No", the execution proceeds to the next layer.



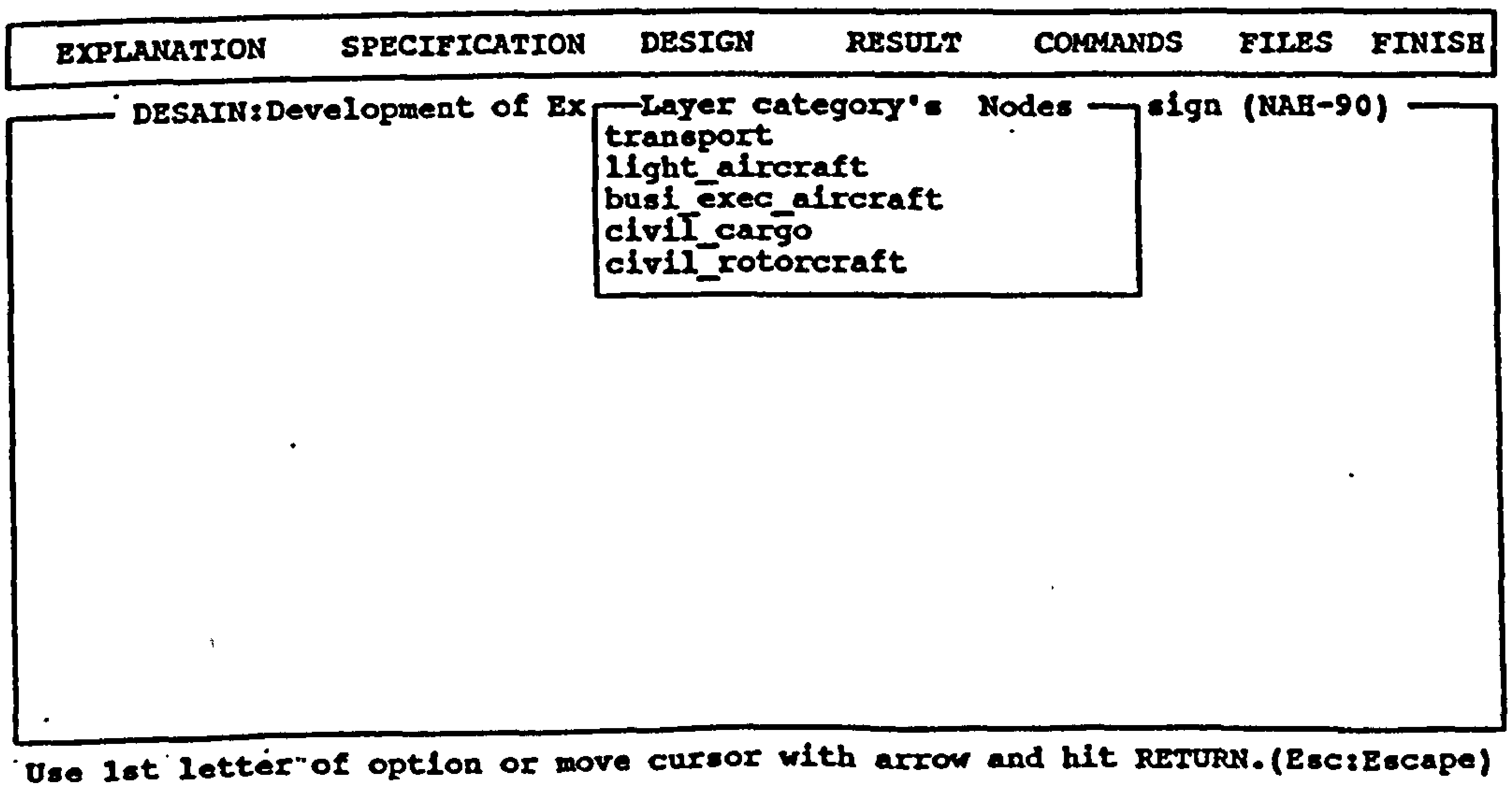
```
DESAIN:Development of Expe  Do you want to backtrack ? (Y/N)
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```

10. If the first layer is implemented, the nodes of next layer "purpose" will be shown the same as the previous layer "vehicle".

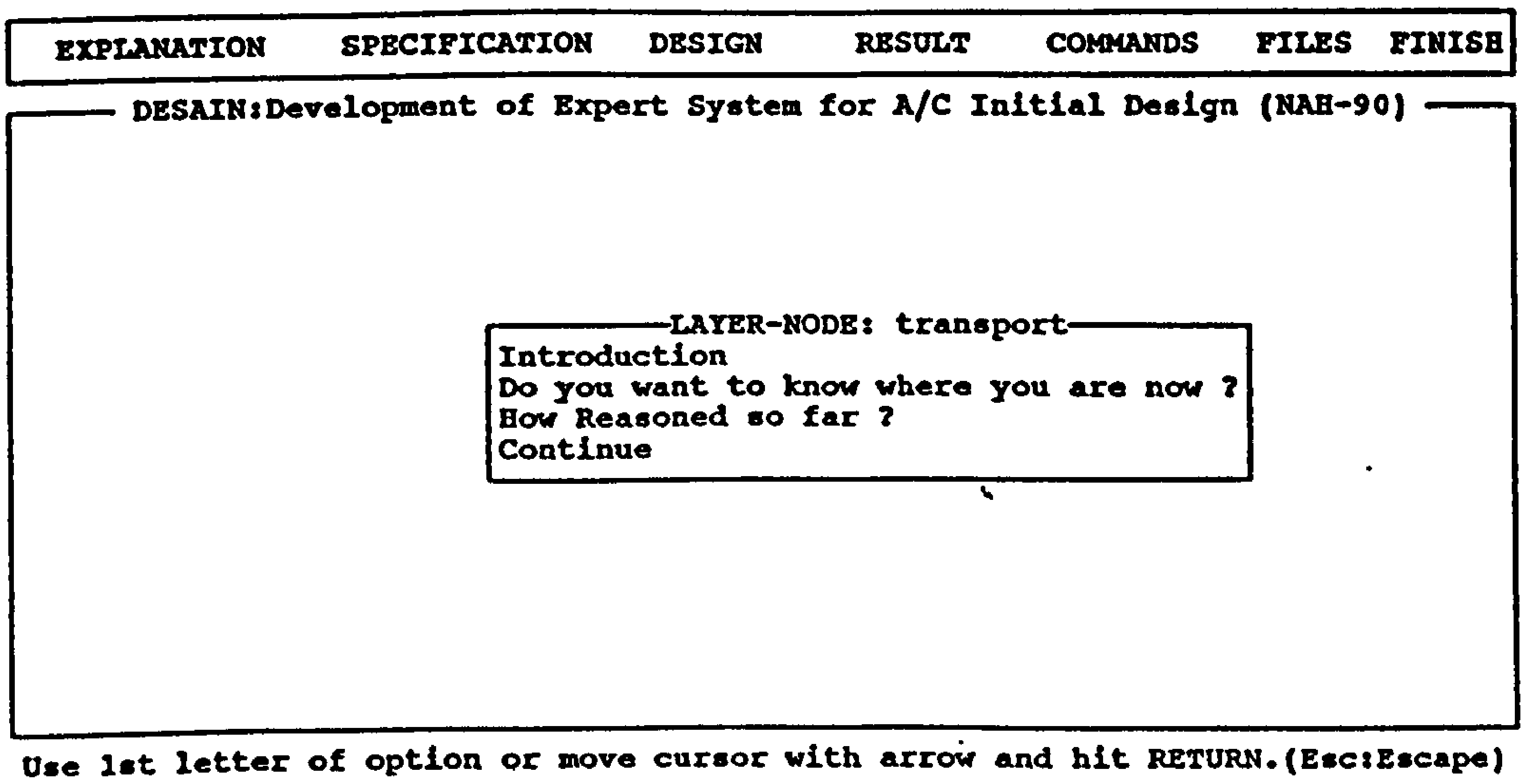


```
DESAIN:Development of Ex  Layer purpose's  Nodes  esign (NAH-90)
                          civil
                          military
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```

11. Likewise, the nodes of such layers as purpose, category, concept, and take-off / land concept will be also shown.



```
DESAIN:Development of Ex  Layer category's  Nodes  sign (NAH-90)
                          transport
                          light_aircraft
                          busi_exec_aircraft
                          civil_cargo
                          civil_rotorcraft
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```

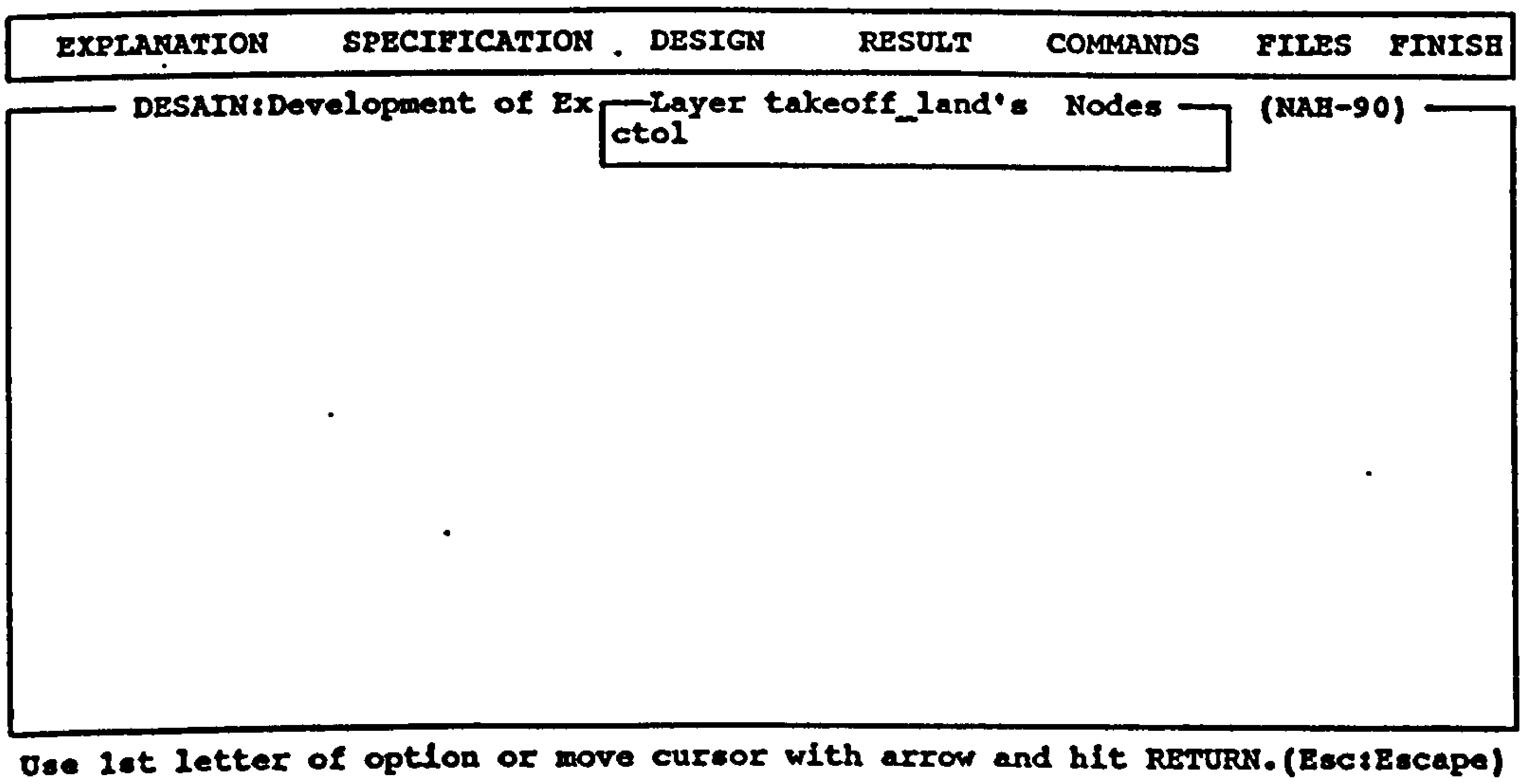


DESAIN:Development of Expert System for A/C Initial Design (NAH-90)

LAYER-NODE: transport
Introduction
Do you want to know where you are now ?
How Reasoned so far ?
Continue

Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)

As explained in the previous chapter, the node appearing on the screen will be constrained by the condition stipulated before this execution. For example, only CTOL appears if the take-off length is over 6000 ft.

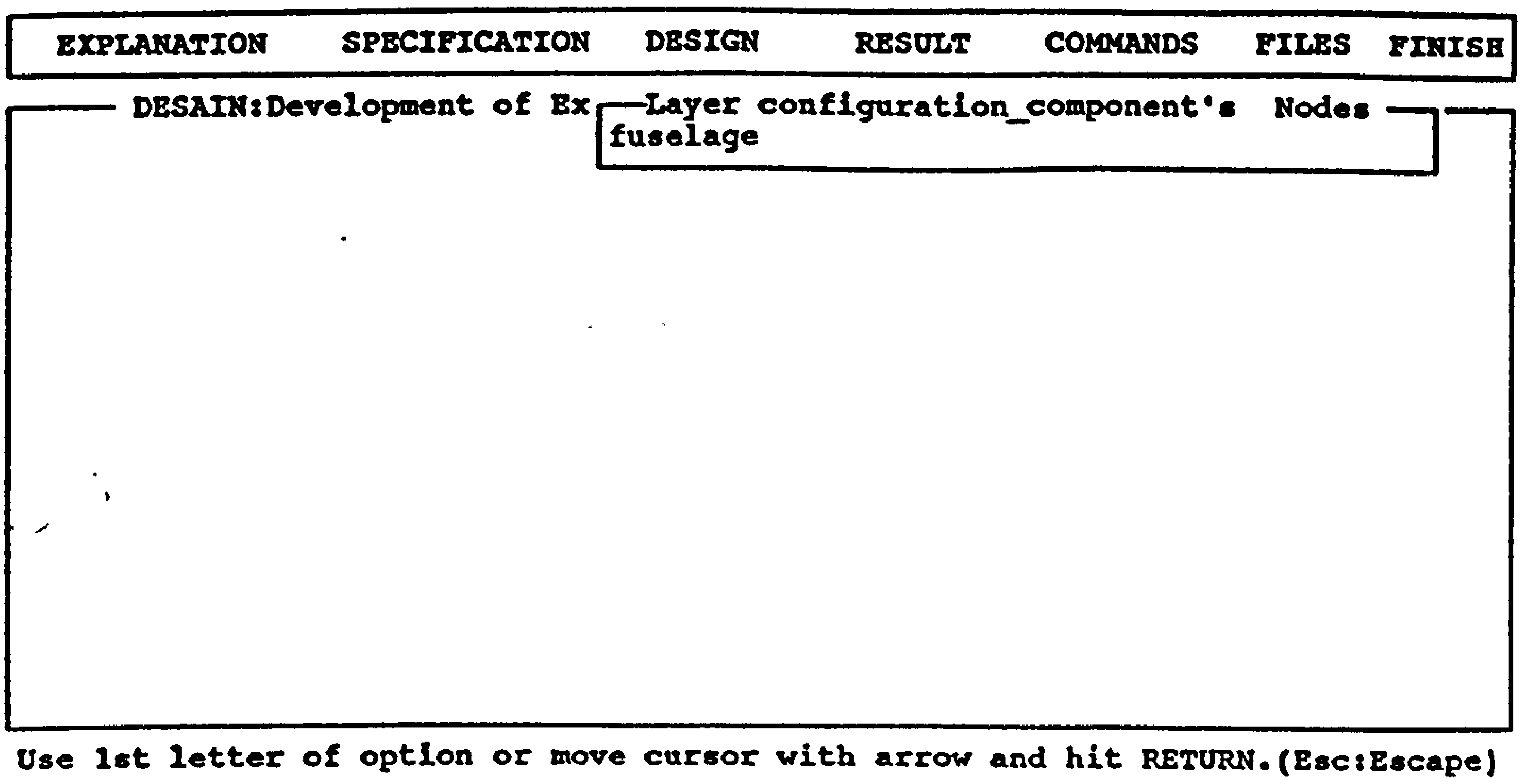


DESAIN:Development of Ex (NAH-90)

Layer takeoff_land's Nodes
ctol

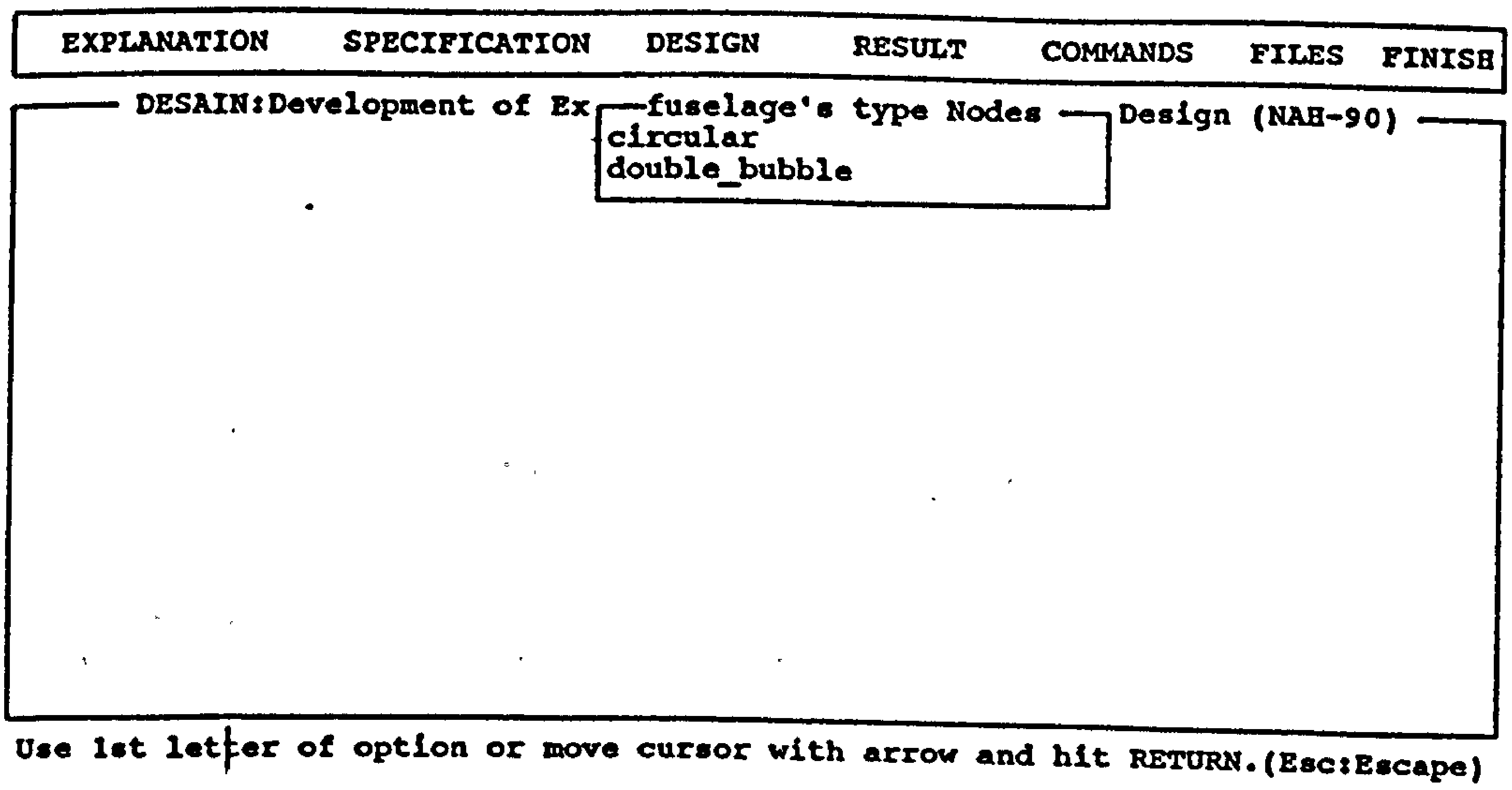
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)

12. After the Set Up phase is implemented, the next phase "Configuration Phase" must be implemented and all its nodes will be shown by the order of precedence. The first node "fuselage" of the layer "Configuration Component" will appear and its sub_layers' (type, number, and position) nodes will be shown for your selection.

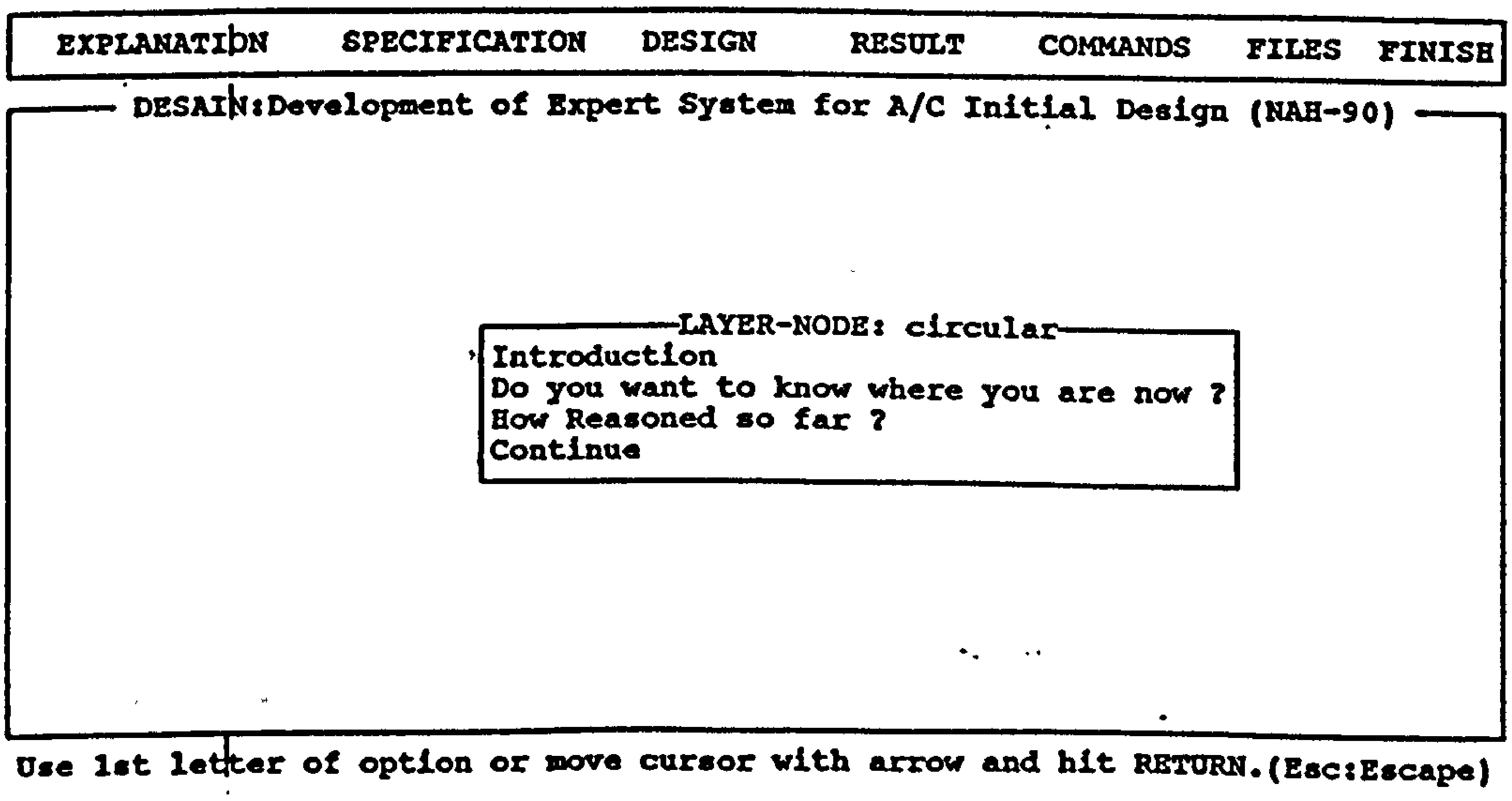


```
DESAIN:Development of Ex
Layer configuration_component's  Nodes
fuselage
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```

The nodes of the sublayer "type", "circular / double_bubble" will be shown first. If you select "circular", the node's items will be shown as previous explanations.

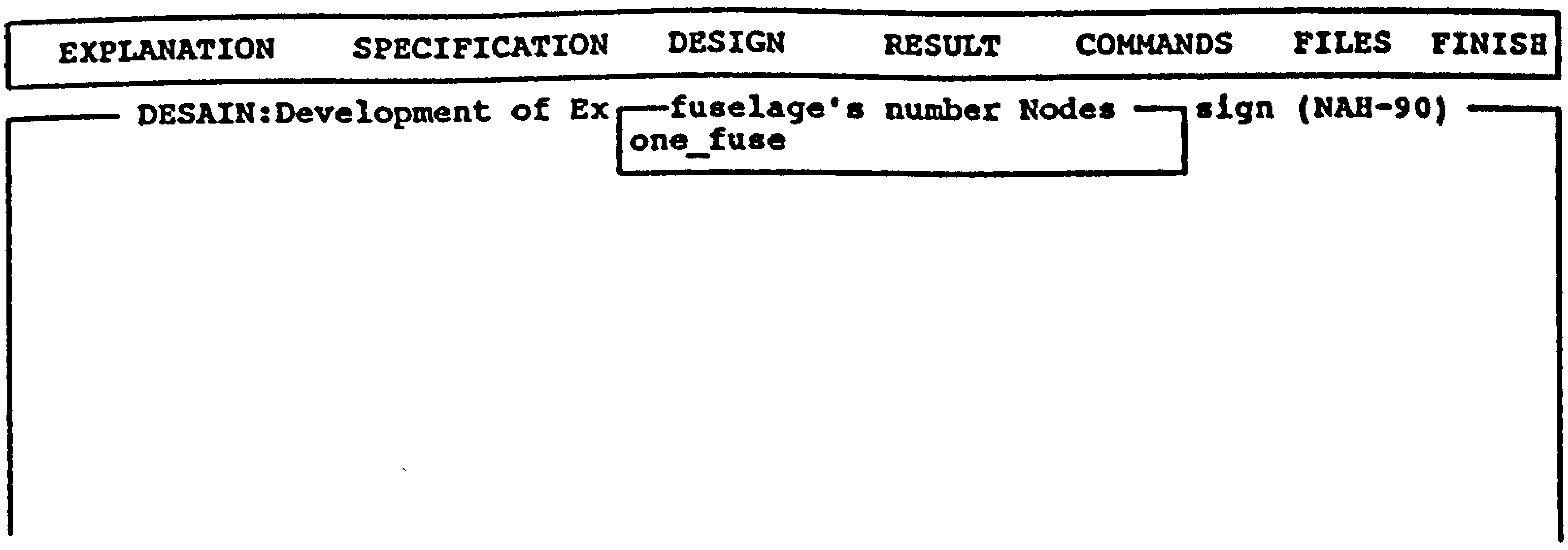


```
DESAIN:Development of Ex
fuselage's type Nodes
circular
double_bubble
Design (NAH-90)
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```

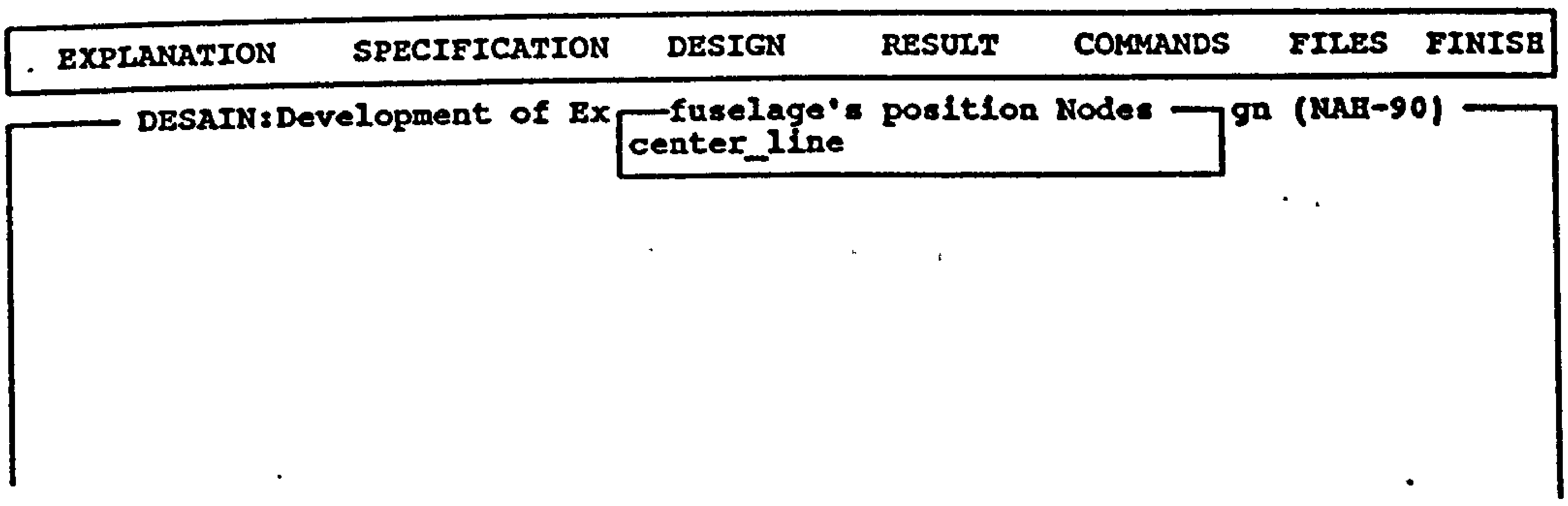


```
DESAIN:Development of Expert System for A/C Initial Design (NAH-90)
LAYER-NODE: circular
Introduction
Do you want to know where you are now ?
How Reasoned so far ?
Continue
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```

13. Then the nodes of sublayers such as number and position will appear.

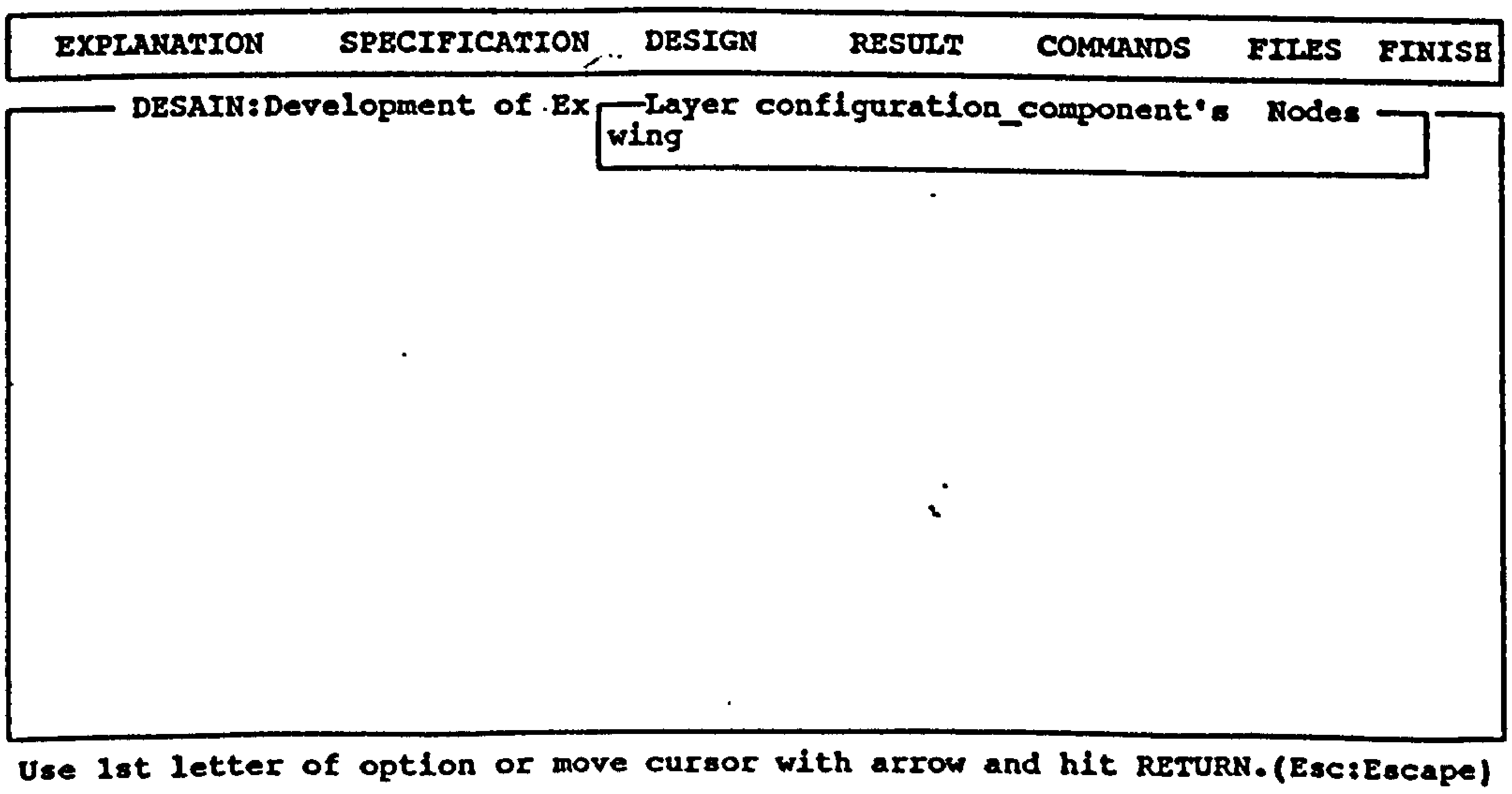


```
DESAIN:Development of Ex —fuseelage's number Nodes — sign (NAH-90)
one_fuse
```



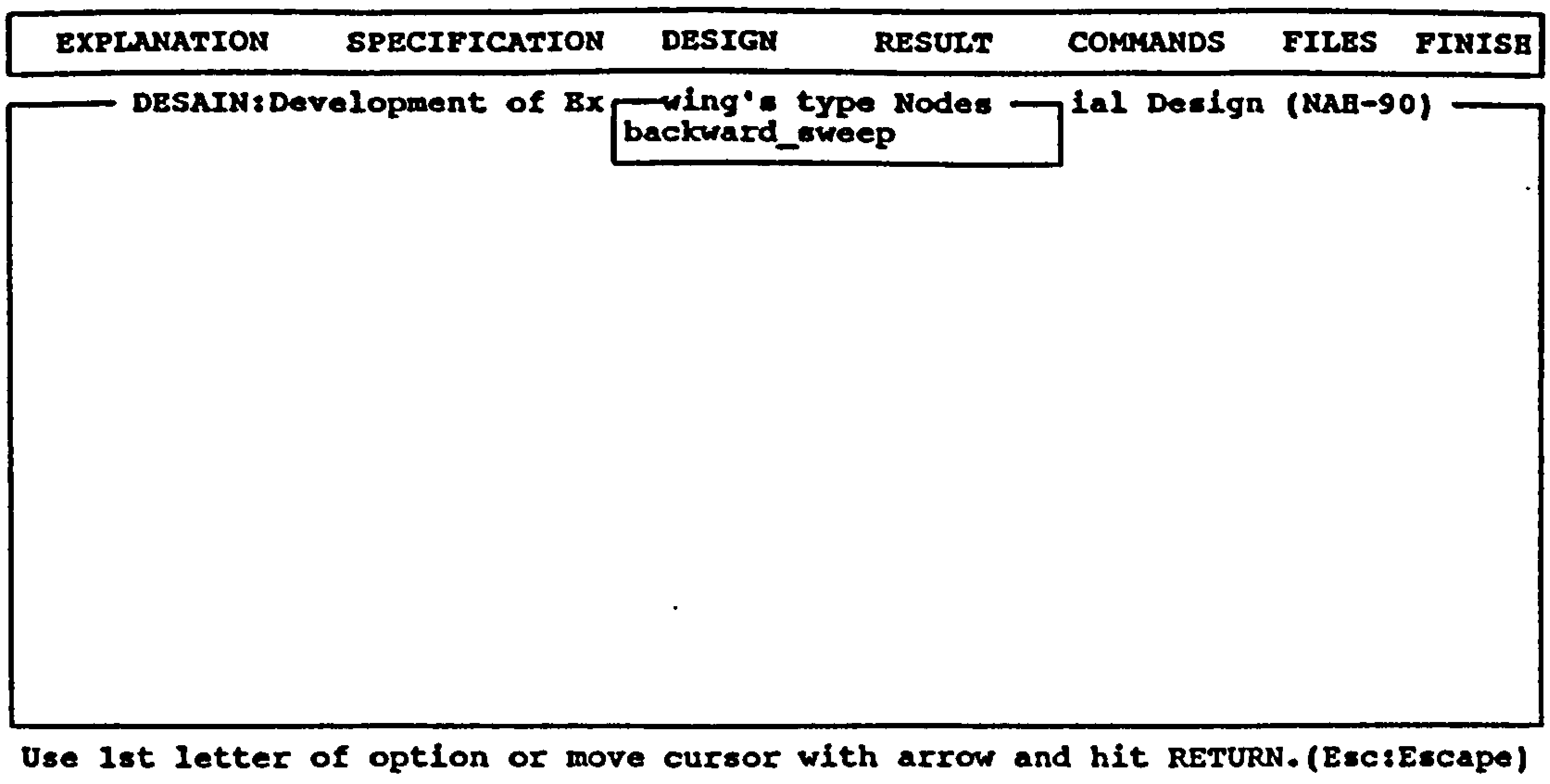
```
DESAIN:Development of Ex —fuselage's position Nodes — gn (NAH-90)
center_line
```

If this implementation finishes, the next main node of the layer "configuration - component" will be executed the same as the "fuselage" implementation by the order of WING, ENGINE, VERTICAL_TAIL, HORIZONTAL_TAIL, and UNDERCARRIAGE.

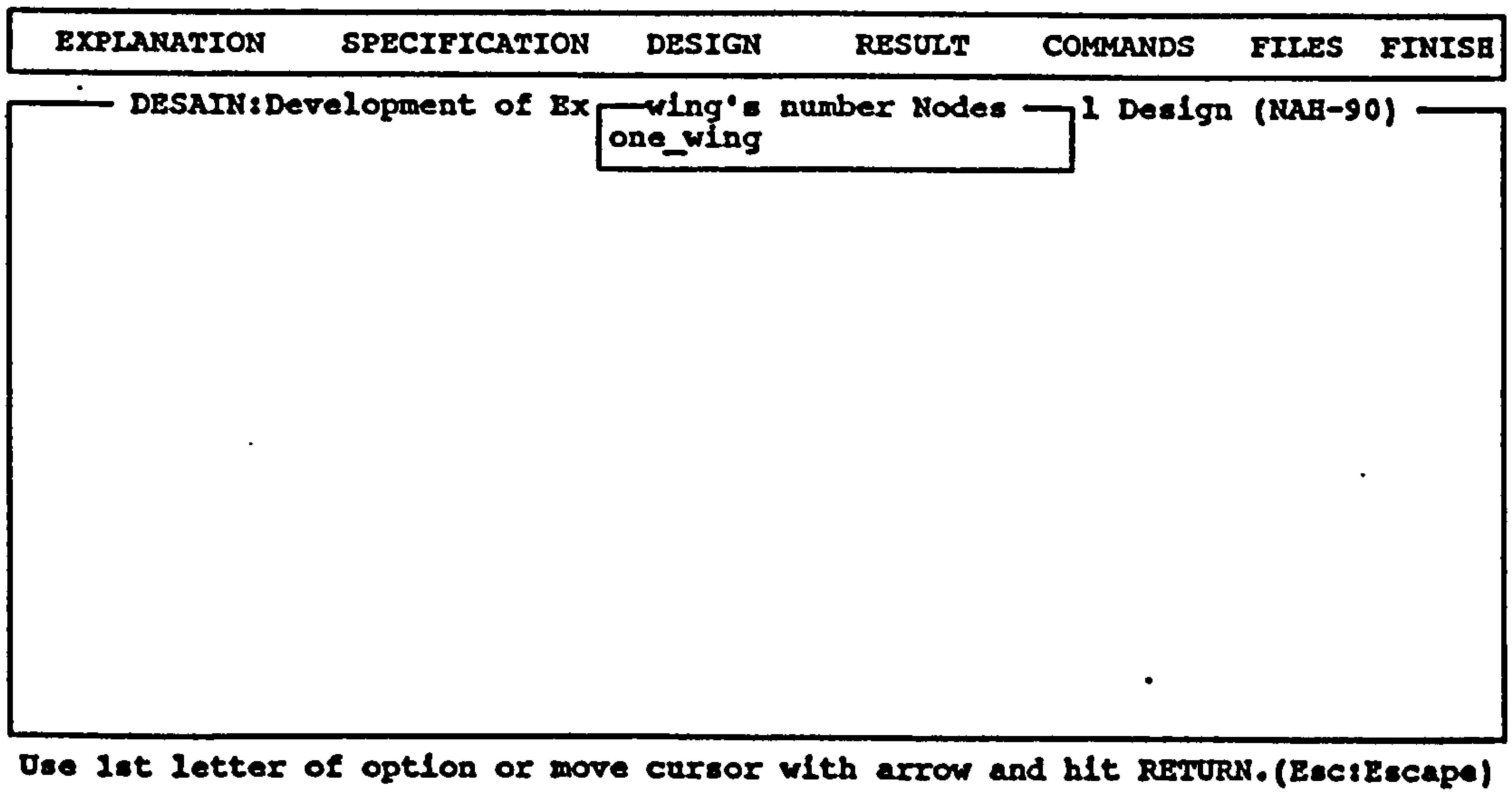


```
DESAIN:Development of Ex —Layer configuration_component's  Nodes —
wing
```

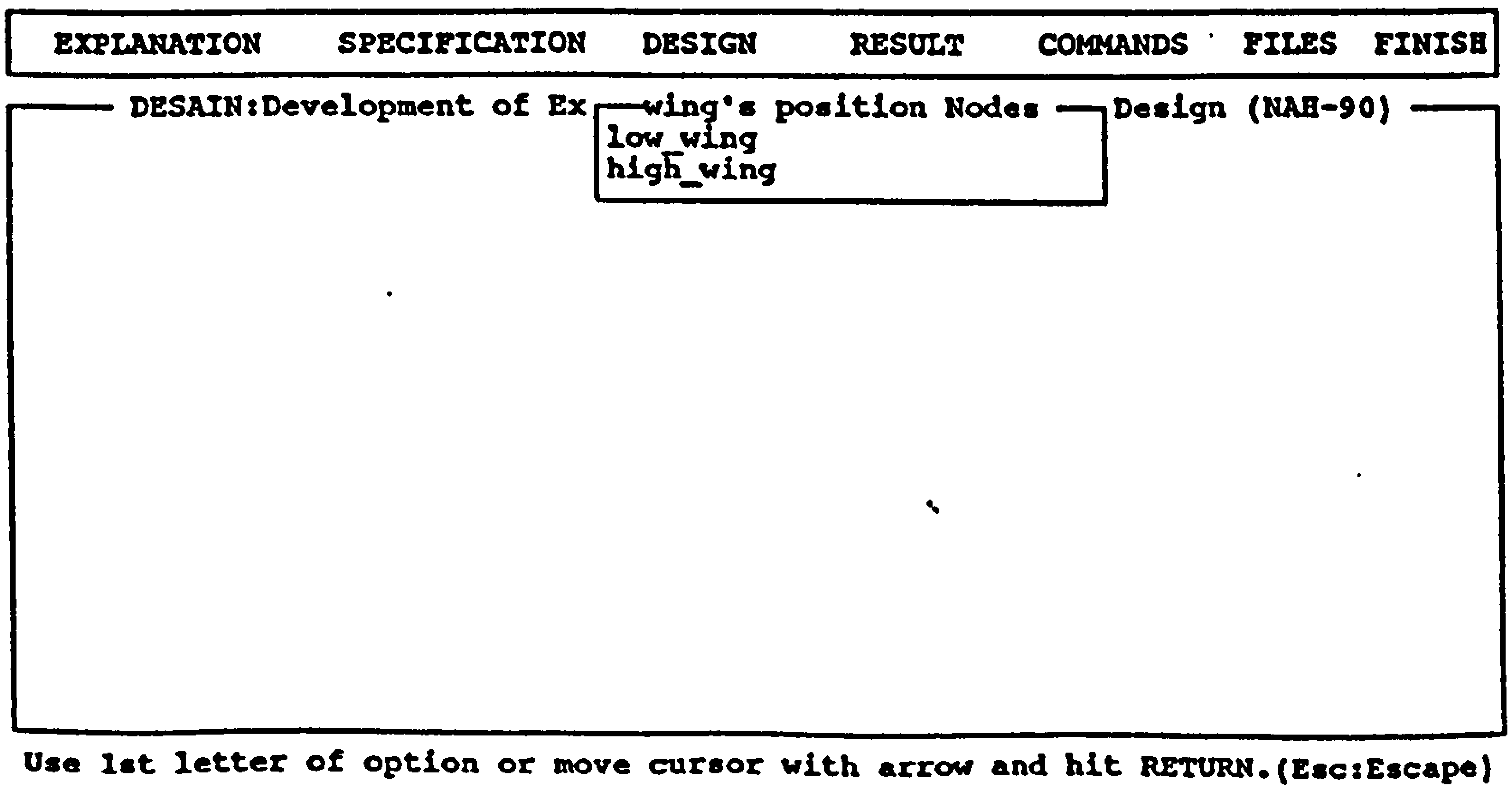
```
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```



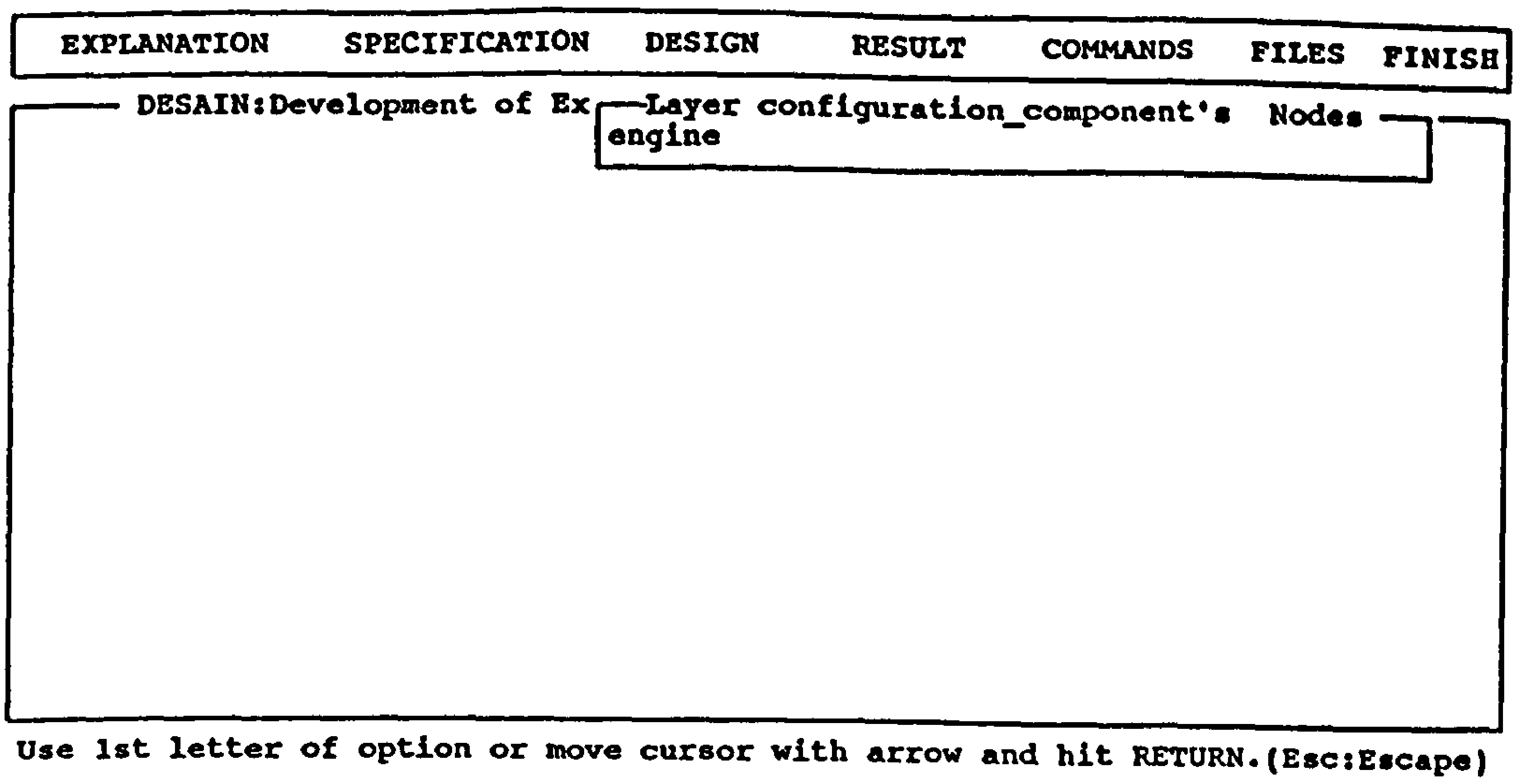
```
DESAIN:Development of Ex
wing's type Nodes
backward_sweep
ial Design (NAH-90)
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```



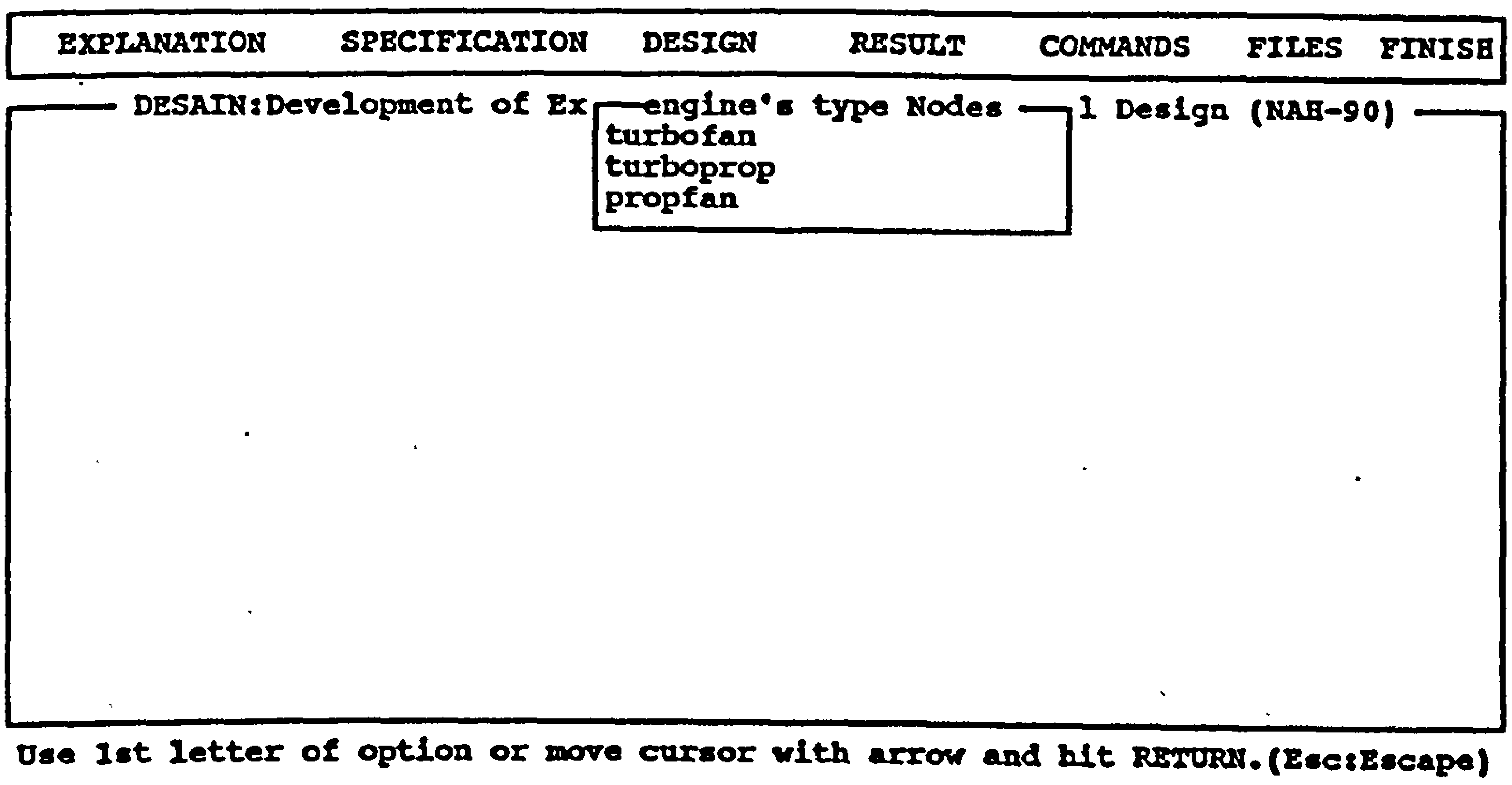
```
DESAIN:Development of Ex
wing's number Nodes
one_wing
l Design (NAH-90)
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```



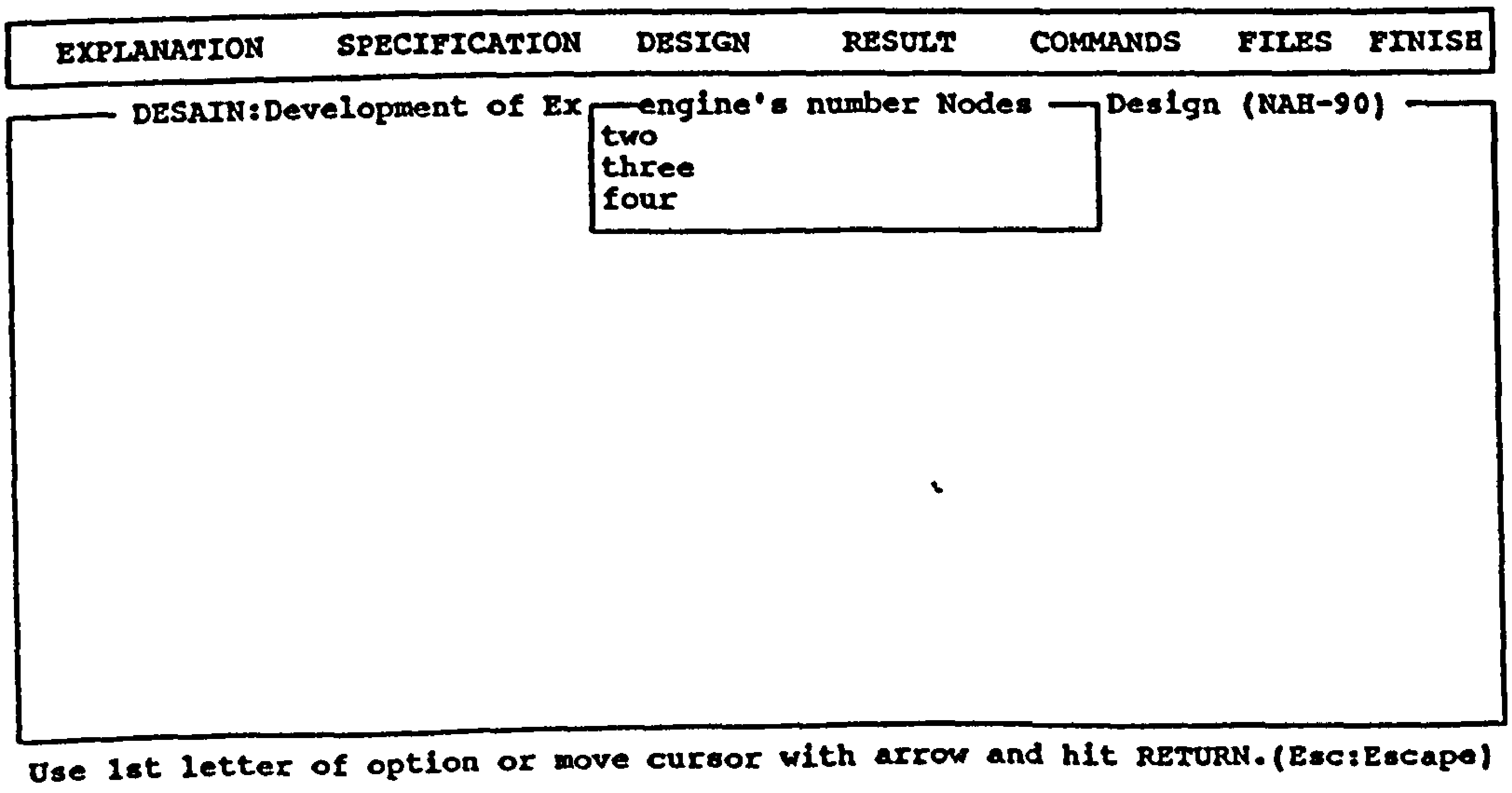
```
DESAIN:Development of Ex
wing's position Nodes
low_wing
high_wing
Design (NAH-90)
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```



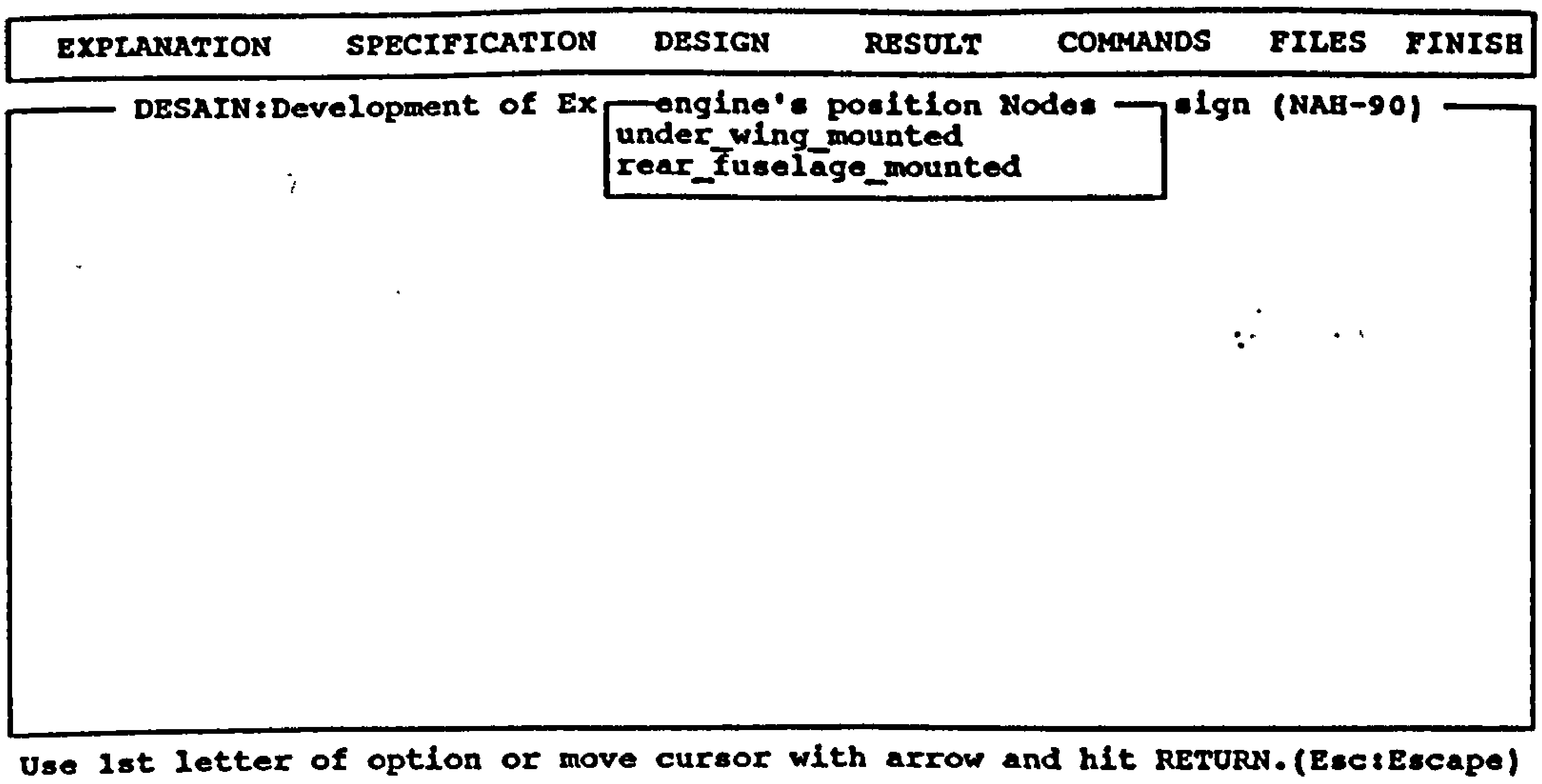
```
EXPLANATION   SPECIFICATION   DESIGN   RESULT   COMMANDS   FILES   FINISH
——— DESAIN:Development of Ex——Layer configuration_component's  Nodes ——
                            engine
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```



```
EXPLANATION   SPECIFICATION   DESIGN   RESULT   COMMANDS   FILES   FINISH
——— DESAIN:Development of Ex——engine's type Nodes ——l Design (NAH-90) ———
                            turbofan
                            turboprop
                            propfan
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```



```
EXPLANATION   SPECIFICATION   DESIGN   RESULT   COMMANDS   FILES   FINISH
——— DESAIN:Development of Ex——engine's number Nodes ——Design (NAH-90) ———
                            two
                            three
                            four
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```



DESAIN:Development of Ex engine's position Nodes sign (NAH-90)

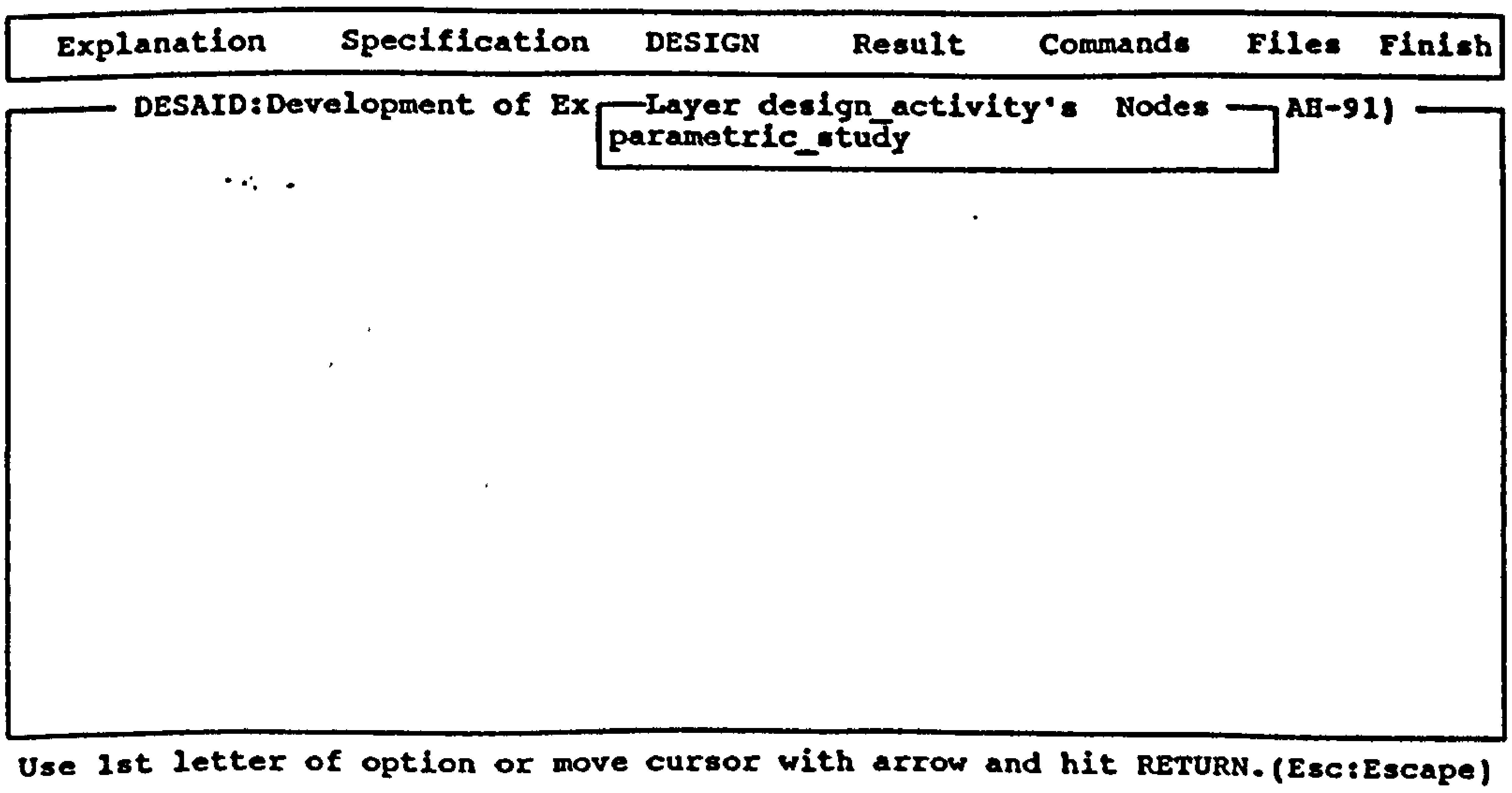
under_wing_mounted
rear_fuselage_mounted

Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)

14. After the configuration phase is implemented, the next phase "Design Phase" will be implemented. The type of this phase is "select_all" and each main node has its sublayers and each sublayer also has its sub_layer nodes.

The main node's sublayer is of the type "select_all", too. The sublayer is here "design_template". The main nodes will be executed in the order of "PARAMETRIC_STUDY, WING_DESIGN, FUSELAGE_DESIGN", and so on. First of all, the main node "PARAMETRIC_STUDY" will be executed.

Thus the nodes of its sub_layer "design_template" will be shown in due order stipulated in the rules.

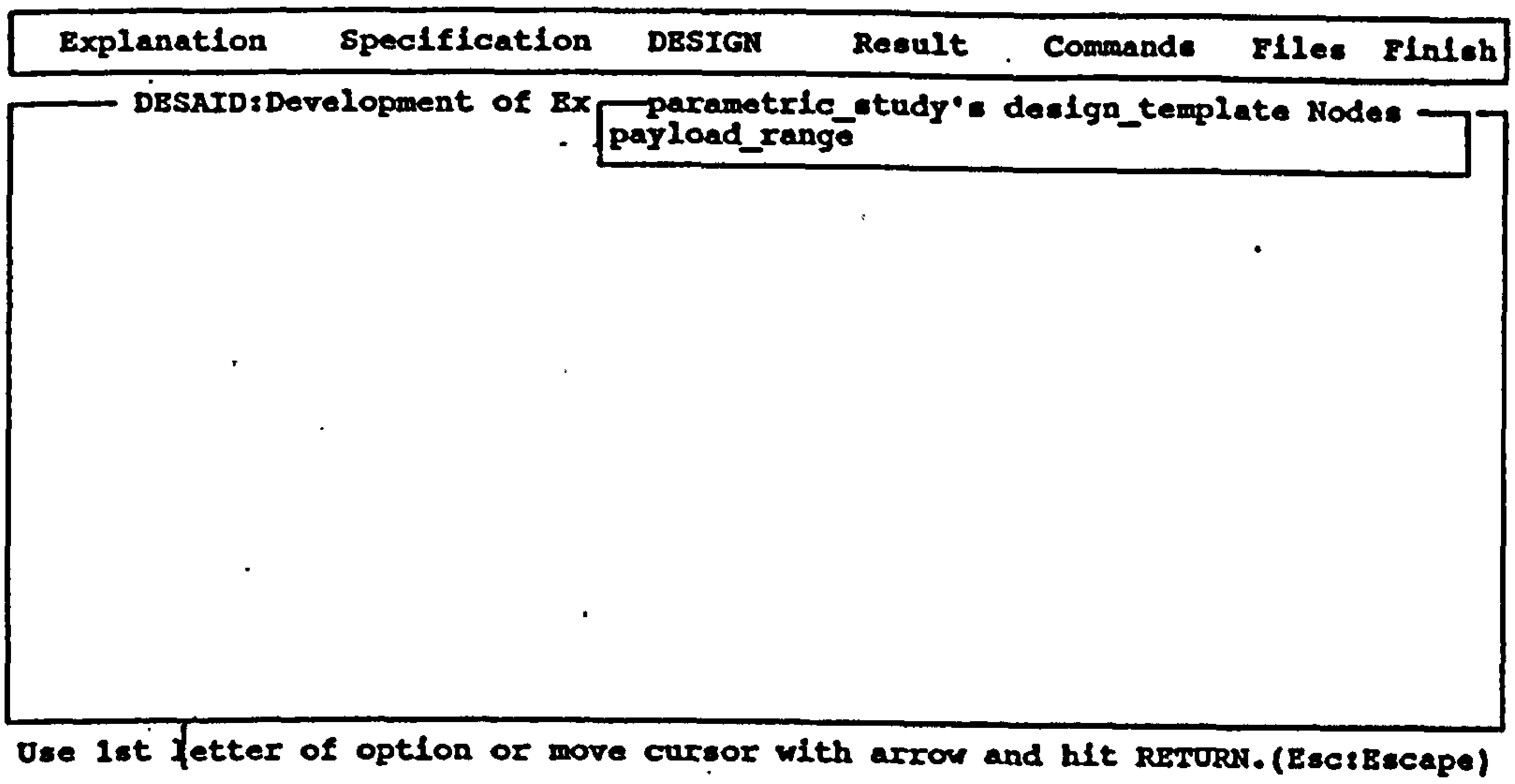


DESAID:Development of Ex Layer design_activity's Nodes AH-91)

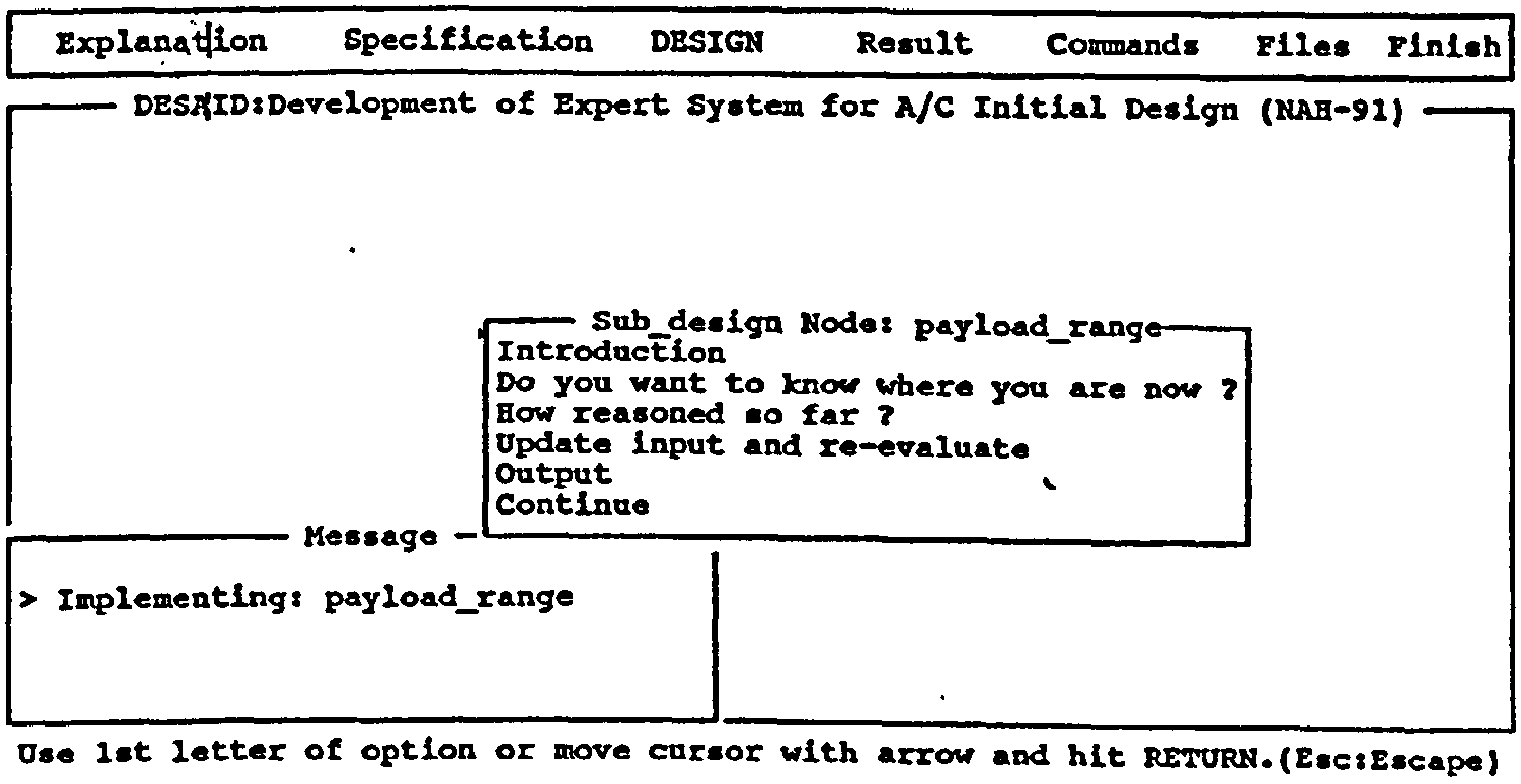
parametric_study

Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)

15. If you select the node "payload_range" of the sublayer "design template" of parametric study, then 6 items will be shown.

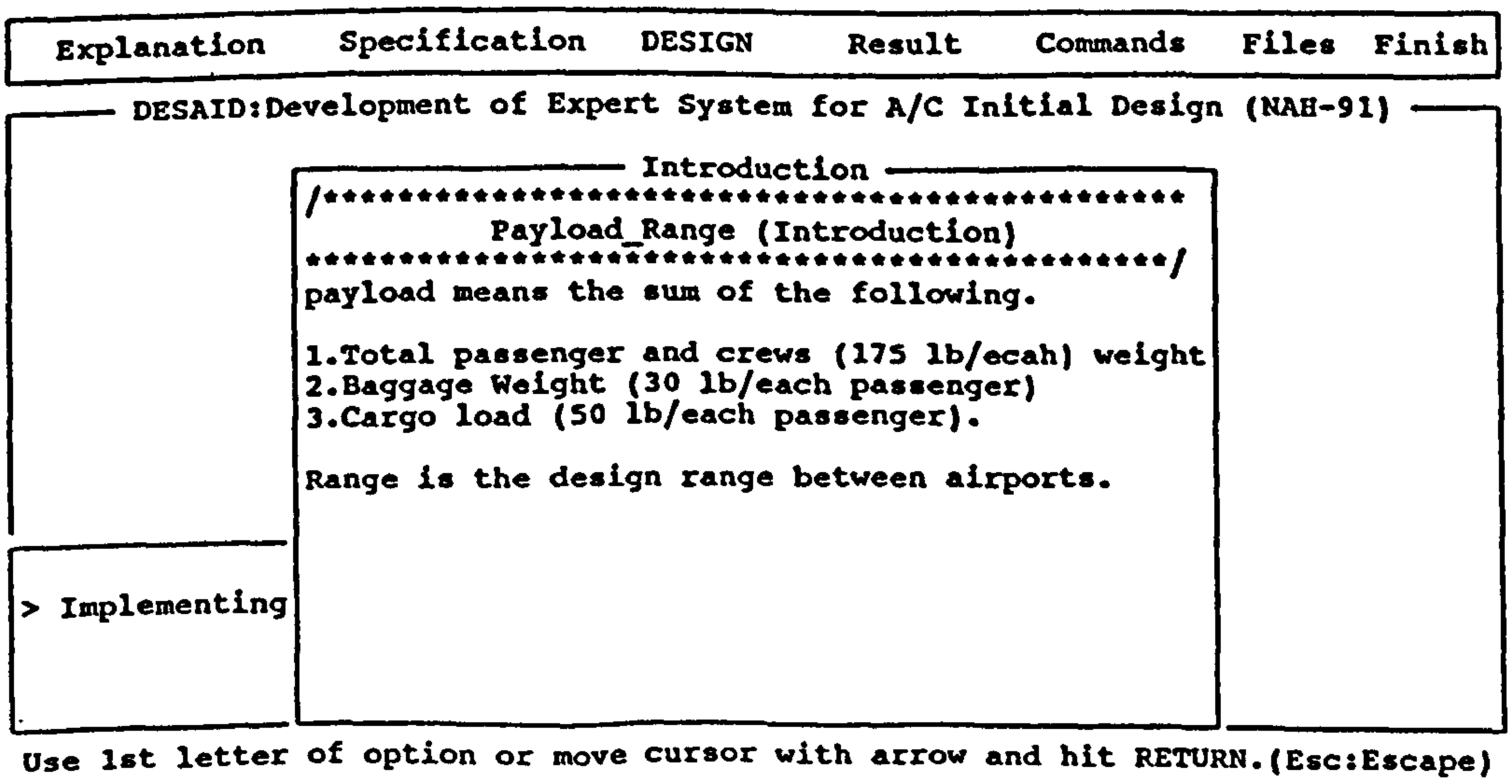


```
DESAID:Development of Ex
parametric_study's design_template Nodes
payload_range
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```



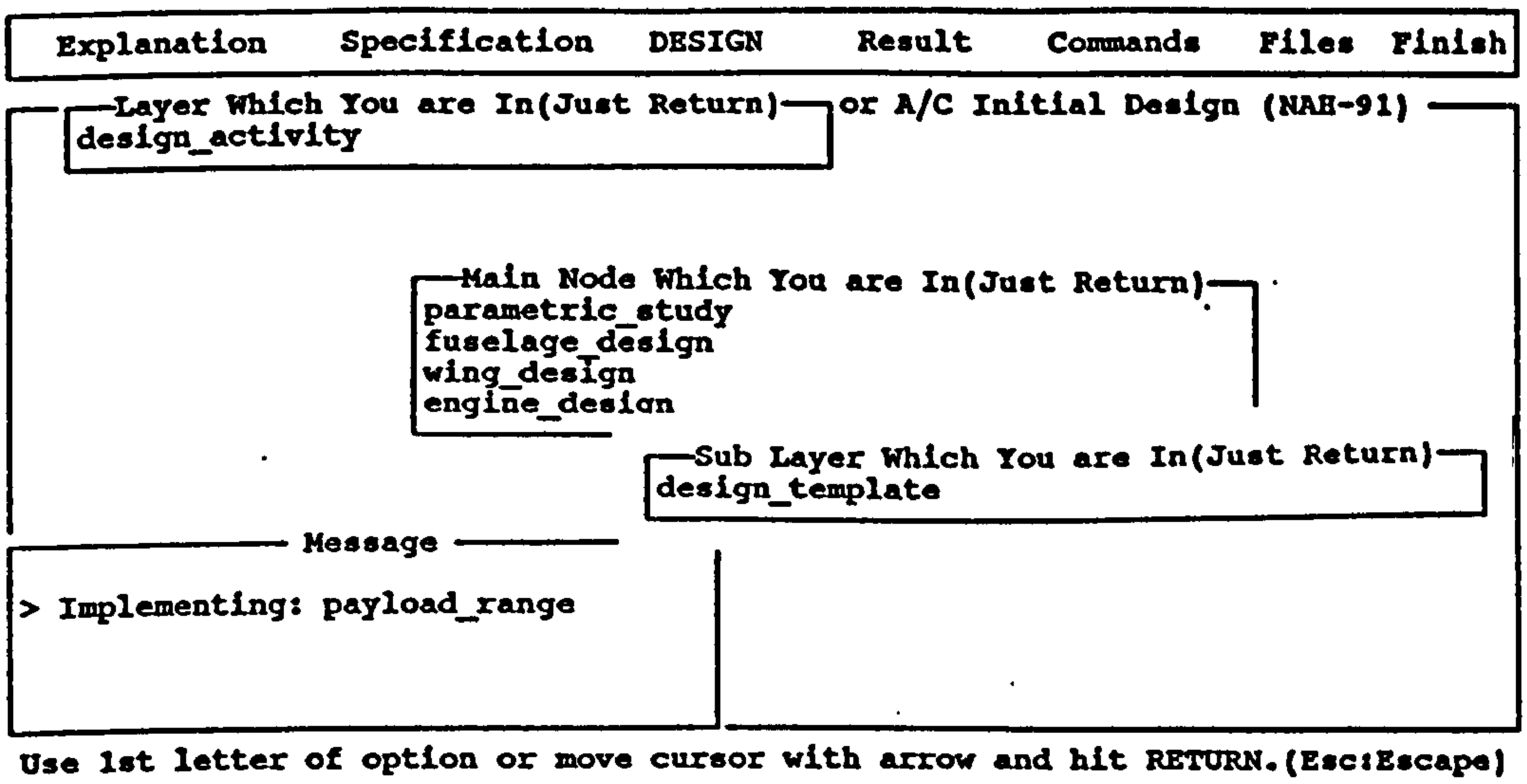
```
DESAID:Development of Expert System for A/C Initial Design (NAH-91)
Sub_design Node: payload_range
Introduction
Do you want to know where you are now ?
How reasoned so far ?
Update input and re-evaluate
Output
Continue
Message
> Implementing: payload_range
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```

The "Introduction" explains the payload range's detail.

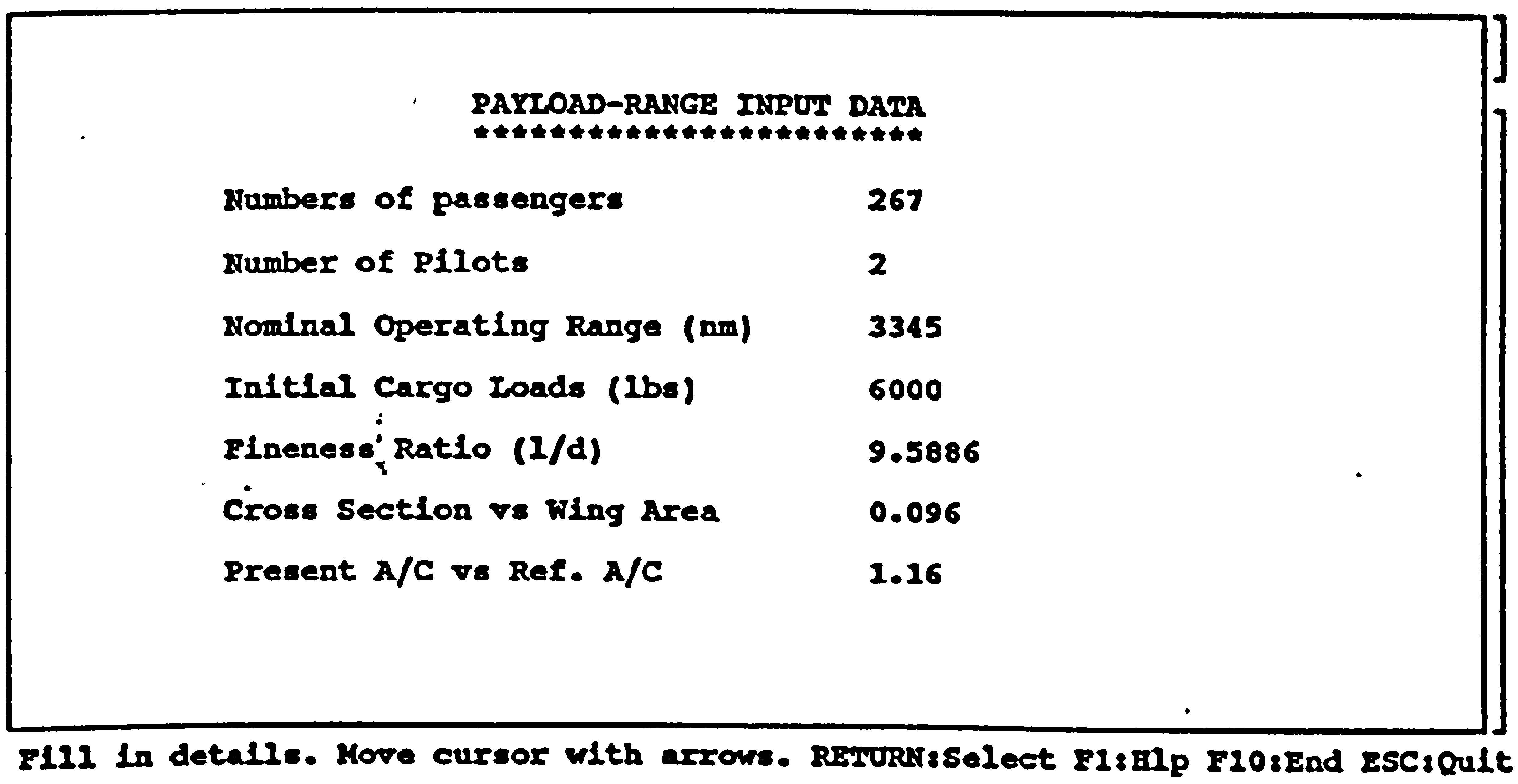


```
DESAID:Development of Expert System for A/C Initial Design (NAH-91)
Introduction
/**********************************************
Payload_Range (Introduction)
**********************************************/
payload means the sum of the following.

1.Total passenger and crews (175 lb/ecah) weight
2.Baggage Weight (30 lb/each passenger)
3.Cargo load (50 lb/each passenger).

Range is the design range between airports.
> Implementing
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```

The item, "Do you want to know where you are now ?" designates the current node of your execution.



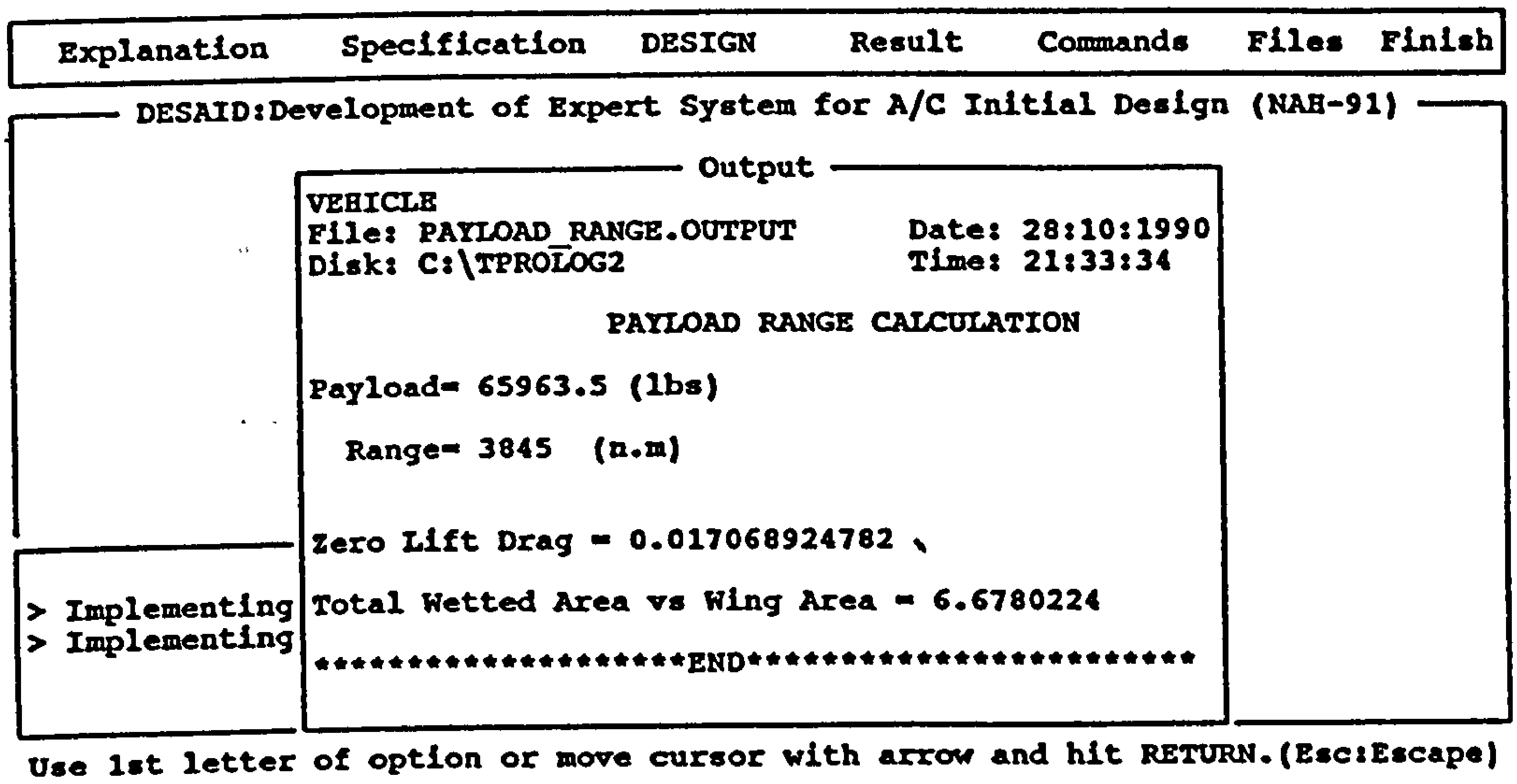
```
Layer Which You are In(Just Return)      or A/C Initial Design (NAH-91)
design_activity

Main Node Which You are In(Just Return)
parametric_study
fuselage_design
wing_design
engine_design

Sub Layer Which You are In(Just Return)
design_template

Message
> Implementing: payload_range

Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```



```
PAYLOAD-RANGE INPUT DATA
************************

Numbers of passengers              267

Number of Pilots                   2

Nominal Operating Range (nm)       3345

Initial Cargo Loads (lbs)          6000

Fineness Ratio (l/d)               9.5886

Cross Section vs Wing Area         0.096

Present A/C vs Ref. A/C            1.16

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

The item "Update Input and Re_evaluate" enables the designer to revise the original input or any values in the specification.

The "Output" shows the execution results in the form of a output file. The detailed analysis of the nodes with respect to input, output, and design analysis is in Appendix I, and their Prolog expressions are in Appendix V.

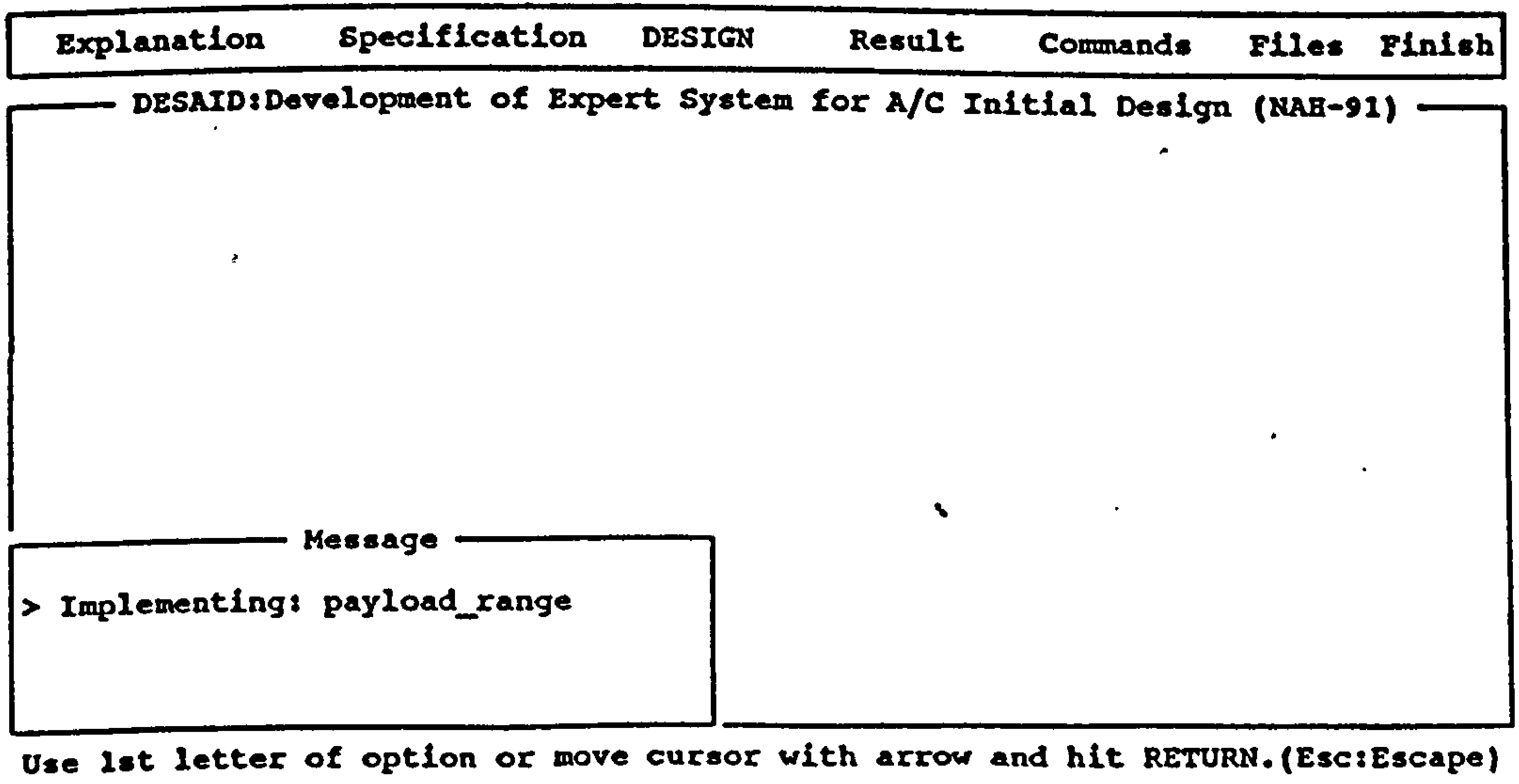


```
VEHICLE
File: PAYLOAD_RANGE.OUTPUT       Date: 28:10:1990
Disk: C:\TPROLOG2                Time: 21:33:34

              PAYLOAD RANGE CALCULATION

Payload= 65963.5 (lbs)

  Range= 3845  (n.m)

Zero Lift Drag = 0.017068924782

Total Wetted Area vs Wing Area = 6.6780224
*******************END***********************
```

```
> Implementing
> Implementing
```

Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)

The "Continue" enables you to proceed to the next step.

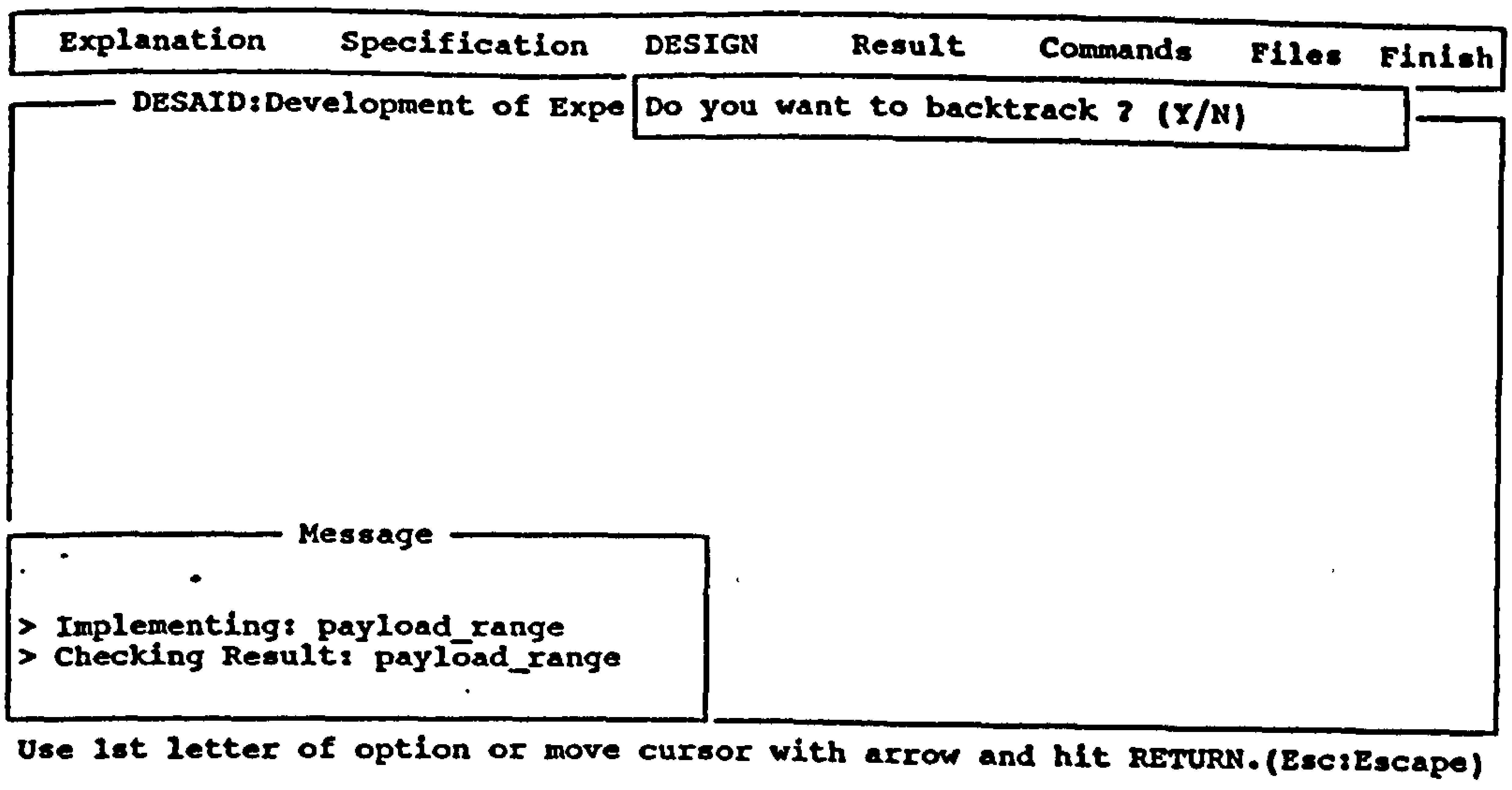


```
> Implementing: payload_range
```

Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)

At this time the system asks your intention whether you want to backtrack or not. If you type "No", the system goes ahead to a next step.

At the same time the system shows the current progress status by showing it in the message window "Implementing" or "Checking the result".



Do you want to backtrack ? (Y/N)

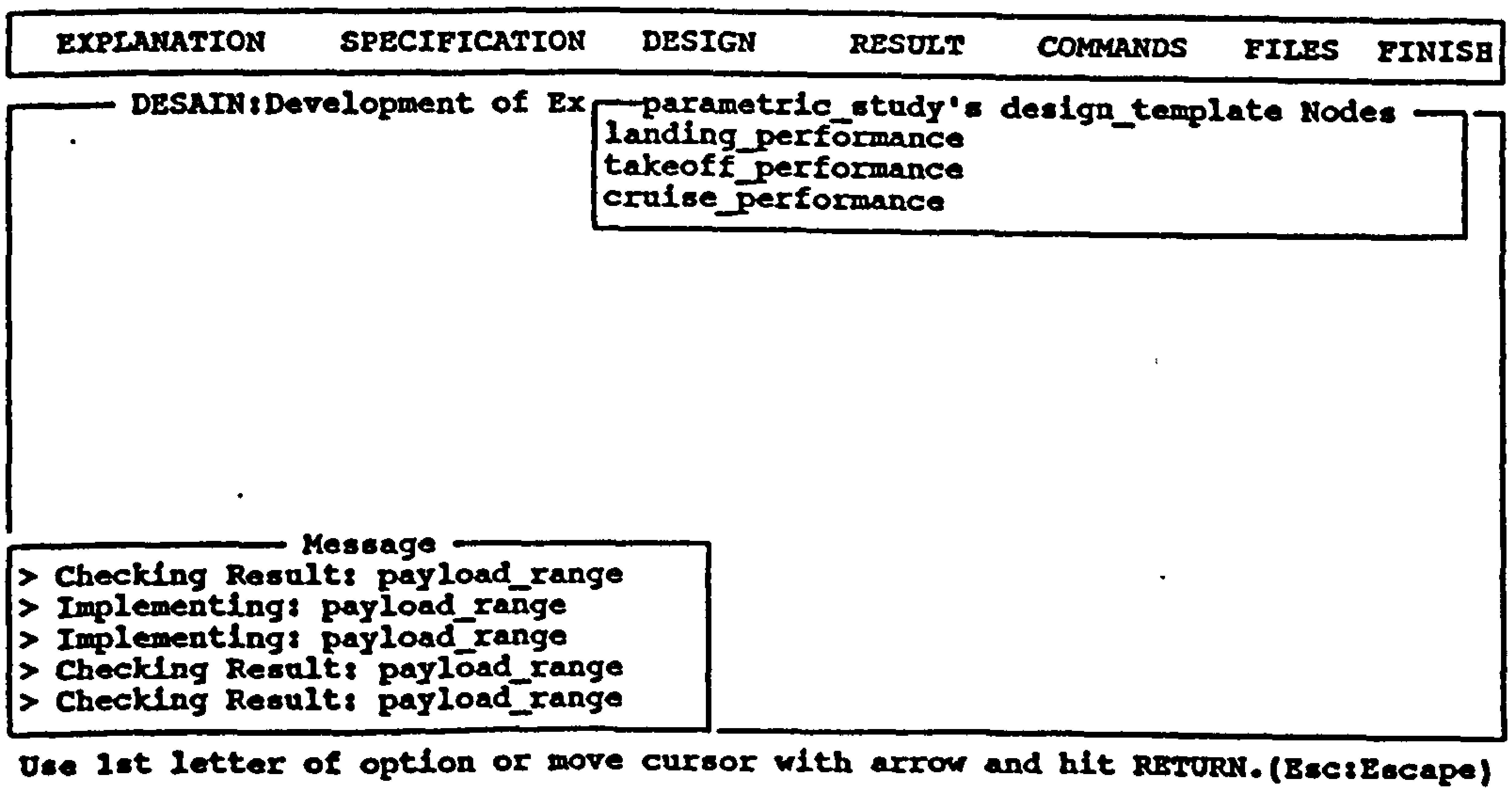
```
> Implementing: payload_range
> Checking Result: payload_range
```

Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)

16. After the payload range is implemented, the next nodes will appear on the screen for your selection, i.e., Landing_Performance, Take_off_Performance, Cruise_Performance.

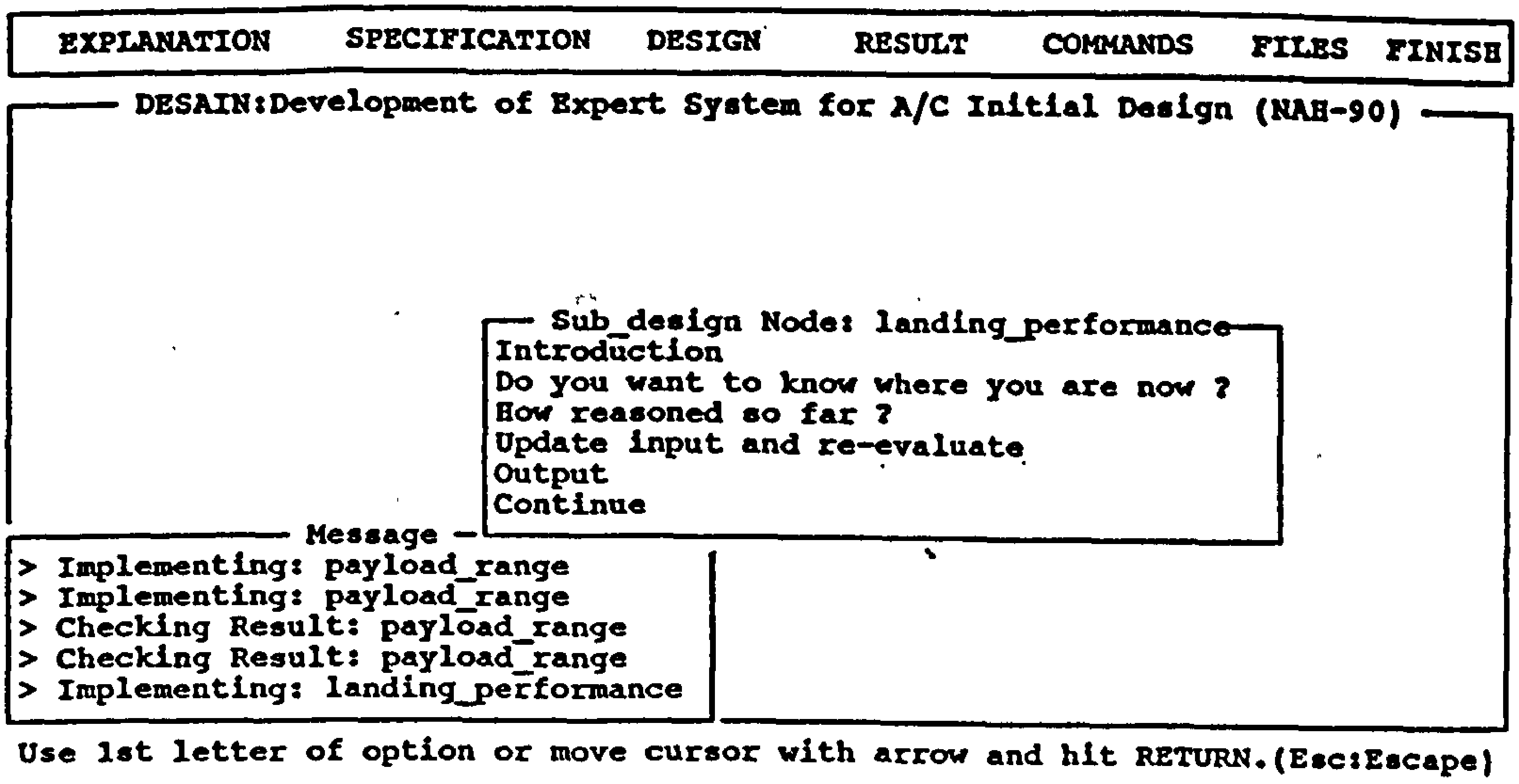
The execution of "Landing_Performance" follows the same procedure as the "Payload_Range".

In this parametric study, the user can get **realistic wing and thrust loadings, gross weight, fuselage diameter, fuselage length, wing area, etc.,** as shown in No. 17 which shows output.



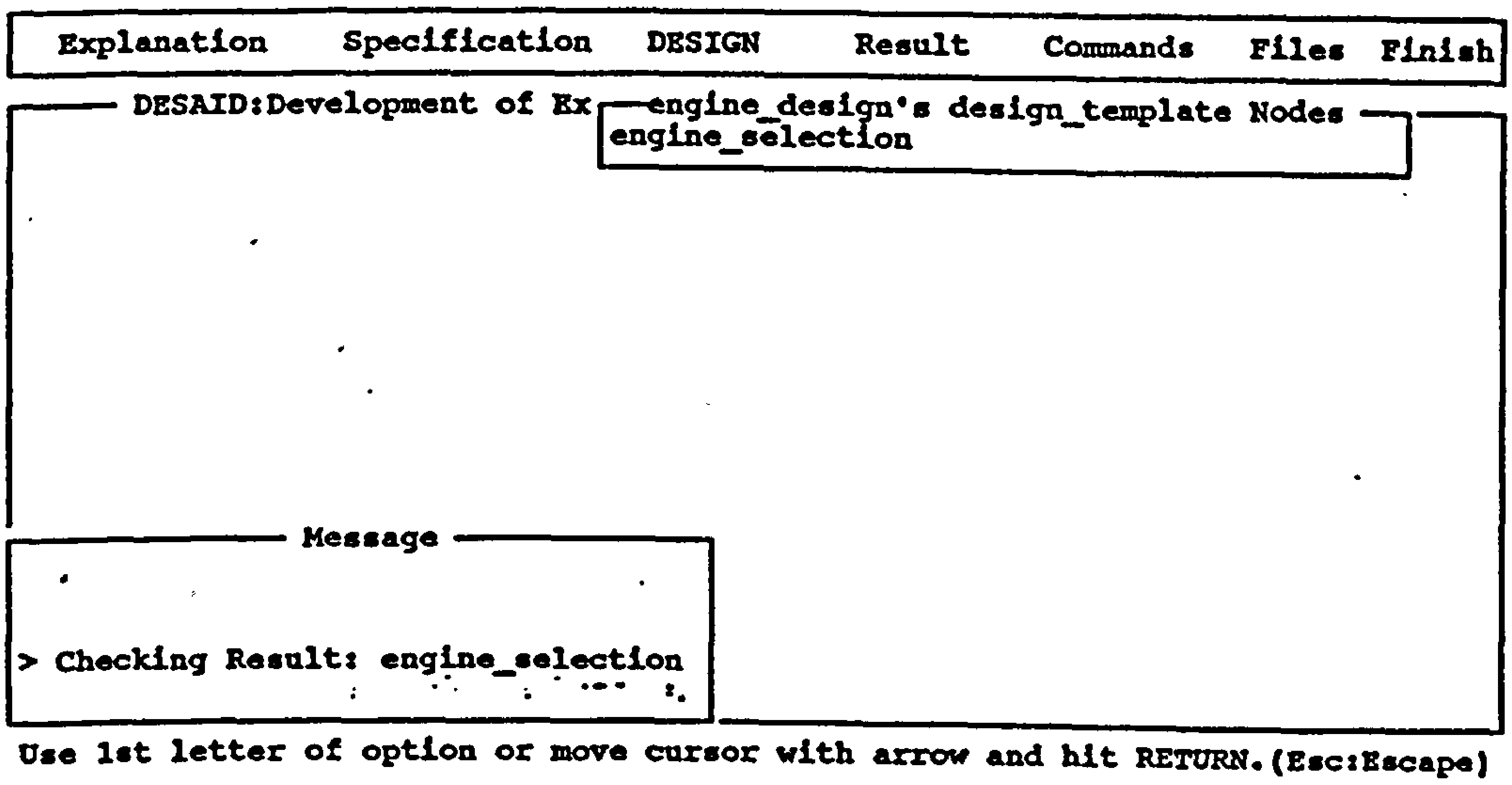
```
> Checking Result: payload_range
> Implementing: payload_range
> Implementing: payload_range
> Checking Result: payload_range
> Checking Result: payload_range
```

Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)



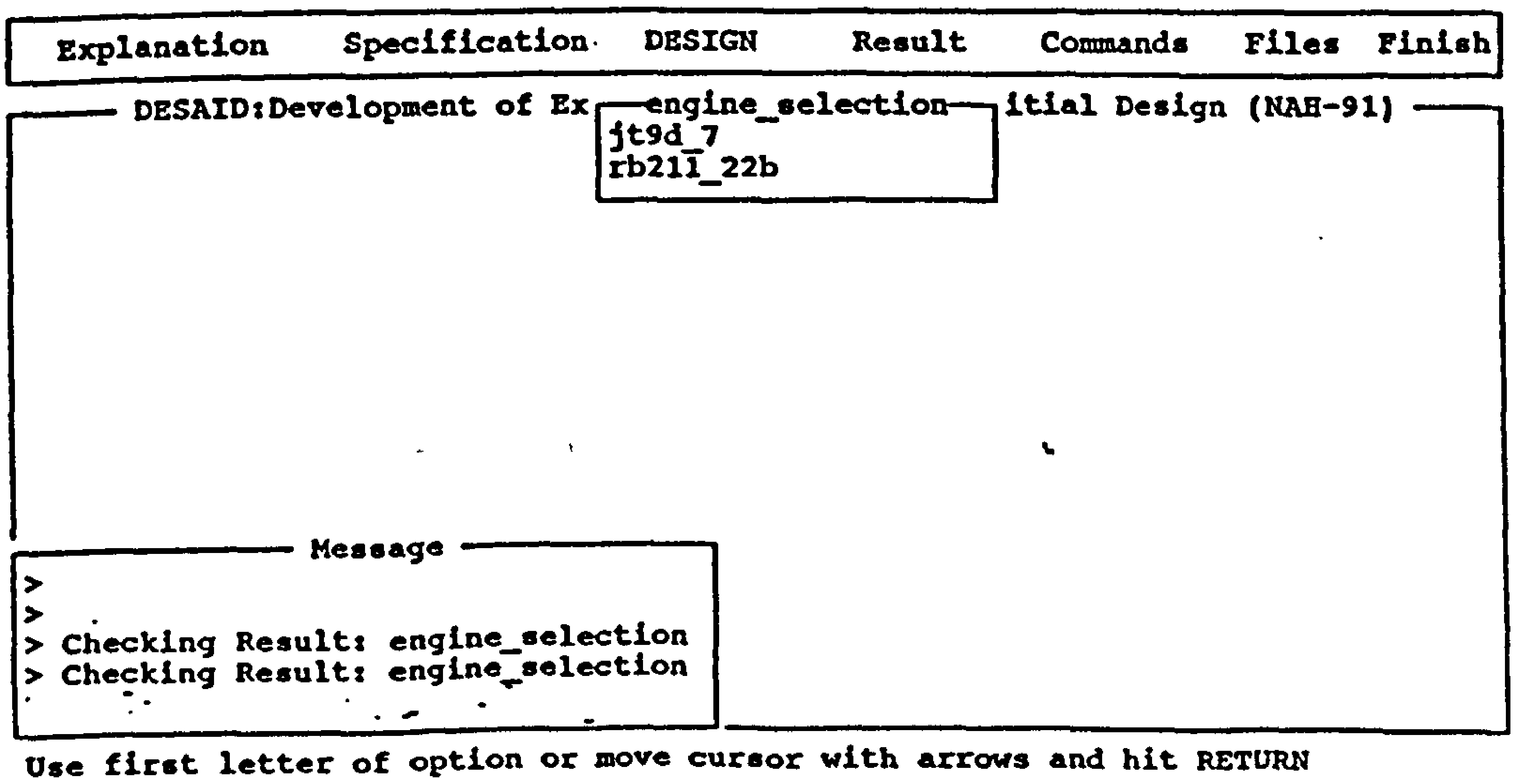
```
> Implementing: payload_range
> Implementing: payload_range
> Checking Result: payload_range
> Checking Result: payload_range
> Implementing: landing_performance
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```

17. Likewise, the wing and fuselage will be done accordingly. Also the "engine selection" will be begun the same procedure as the parametric study.



```
> Checking Result: engine_selection
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```

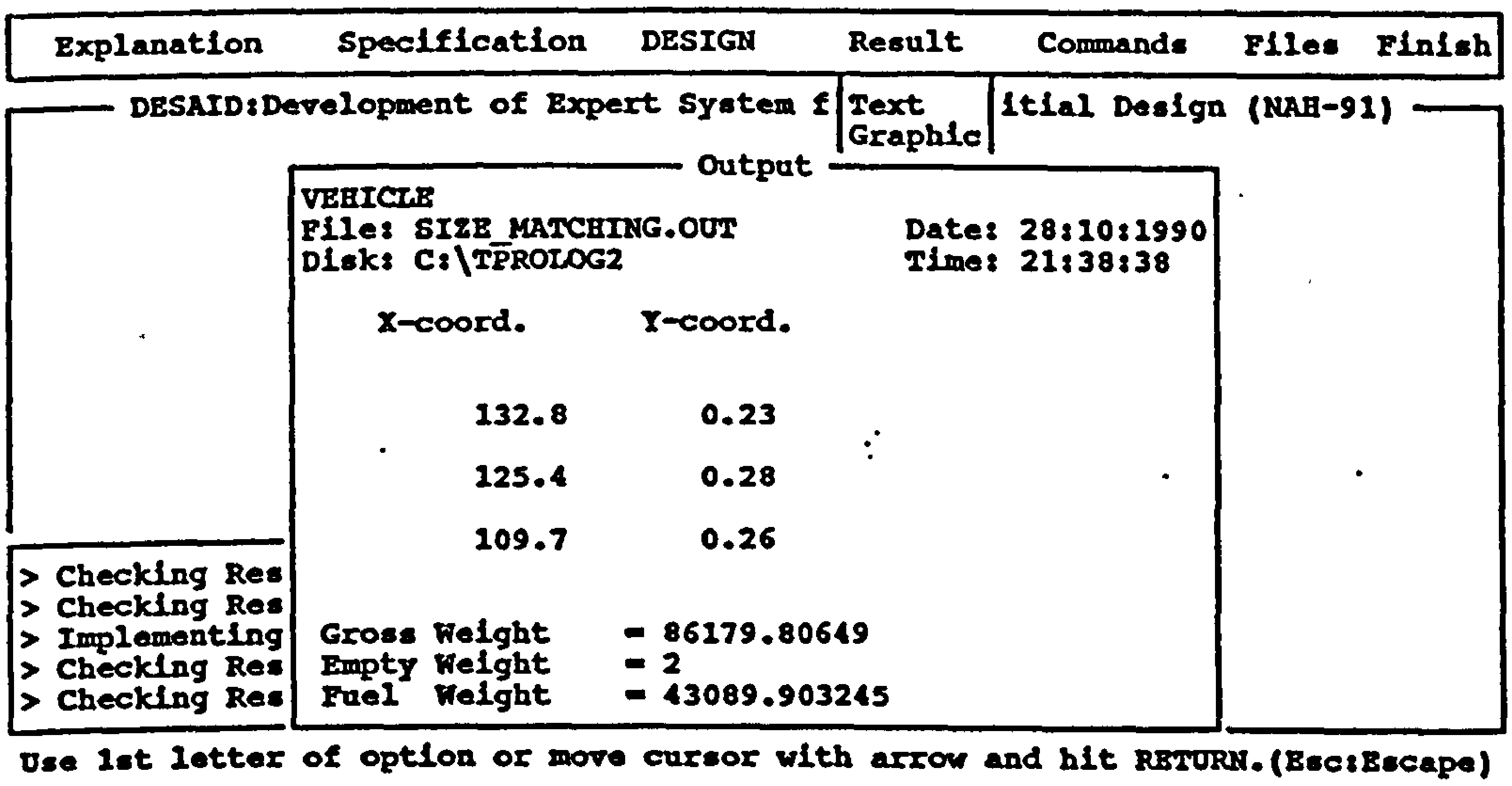
In engine design, the realistic engine thrust is calculated and its output such as real engine, numbers, weight, etc., can be shown. The calculation procedure and its prolog expressions are in Appendix I and V, respectively.



```
>
>
> Checking Result: engine_selection
> Checking Result: engine_selection
Use first letter of option or move cursor with arrows and hit RETURN
```

18. If you go down to the "Result" pull down menu, the required items will be shown. If you select "Text", the system shows all the implemented nodes for your choice. For example, if you select the parametric study, the system will show its output file.

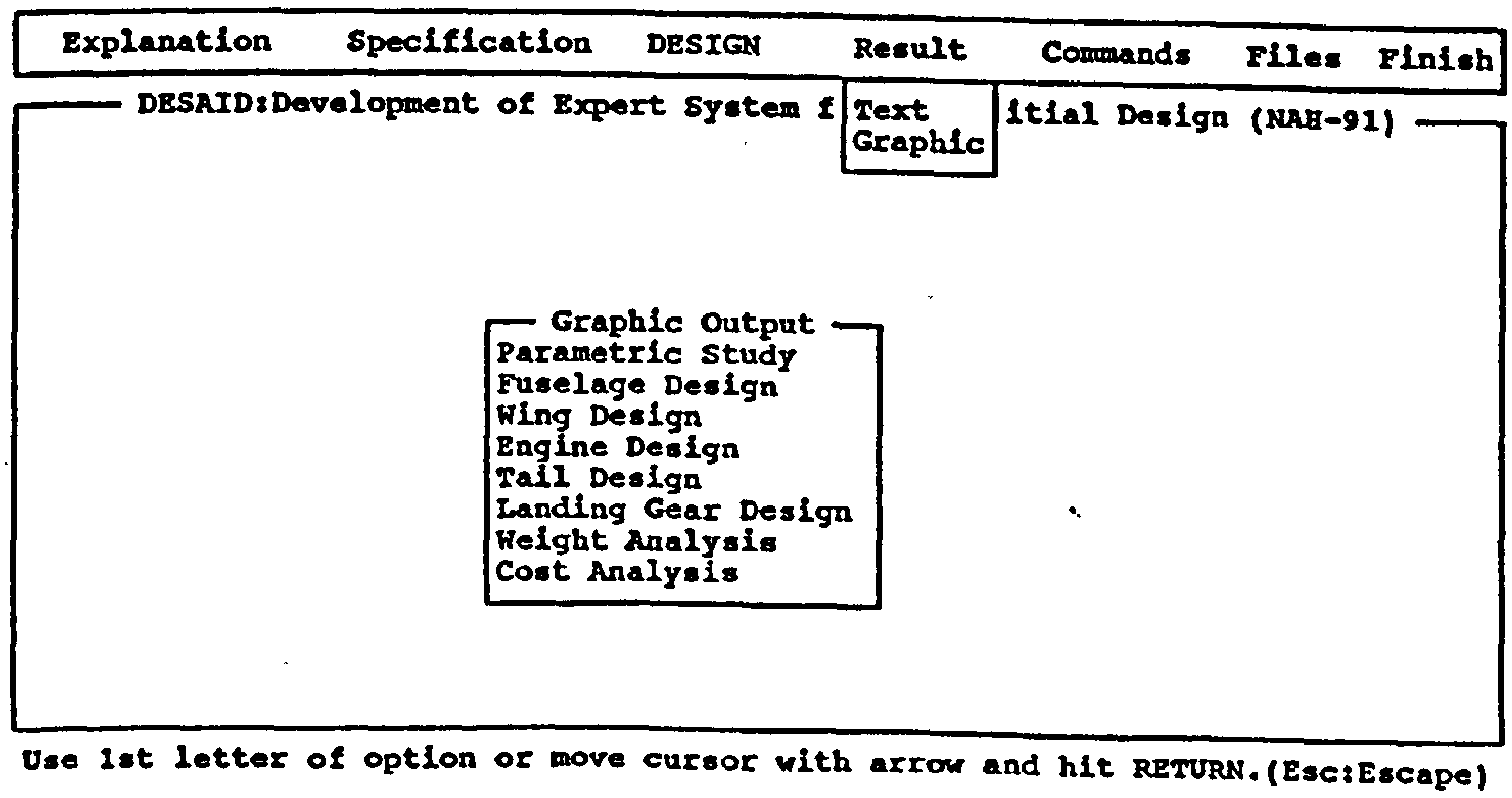
As explained in chapter 4 and Appendix V, the designer can get the required output as shown in the following screen.



```
VEHICLE
File: SIZE_MATCHING.OUT                 Date: 28:10:1990
Disk: C:\TPROLOG2                       Time: 21:38:38

   X-coord.      Y-coord.


        132.8        0.23

        125.4        0.28

        109.7        0.26


Gross Weight     = 86179.80649
Empty Weight     = 2
Fuel  Weight     = 43089.903245
```

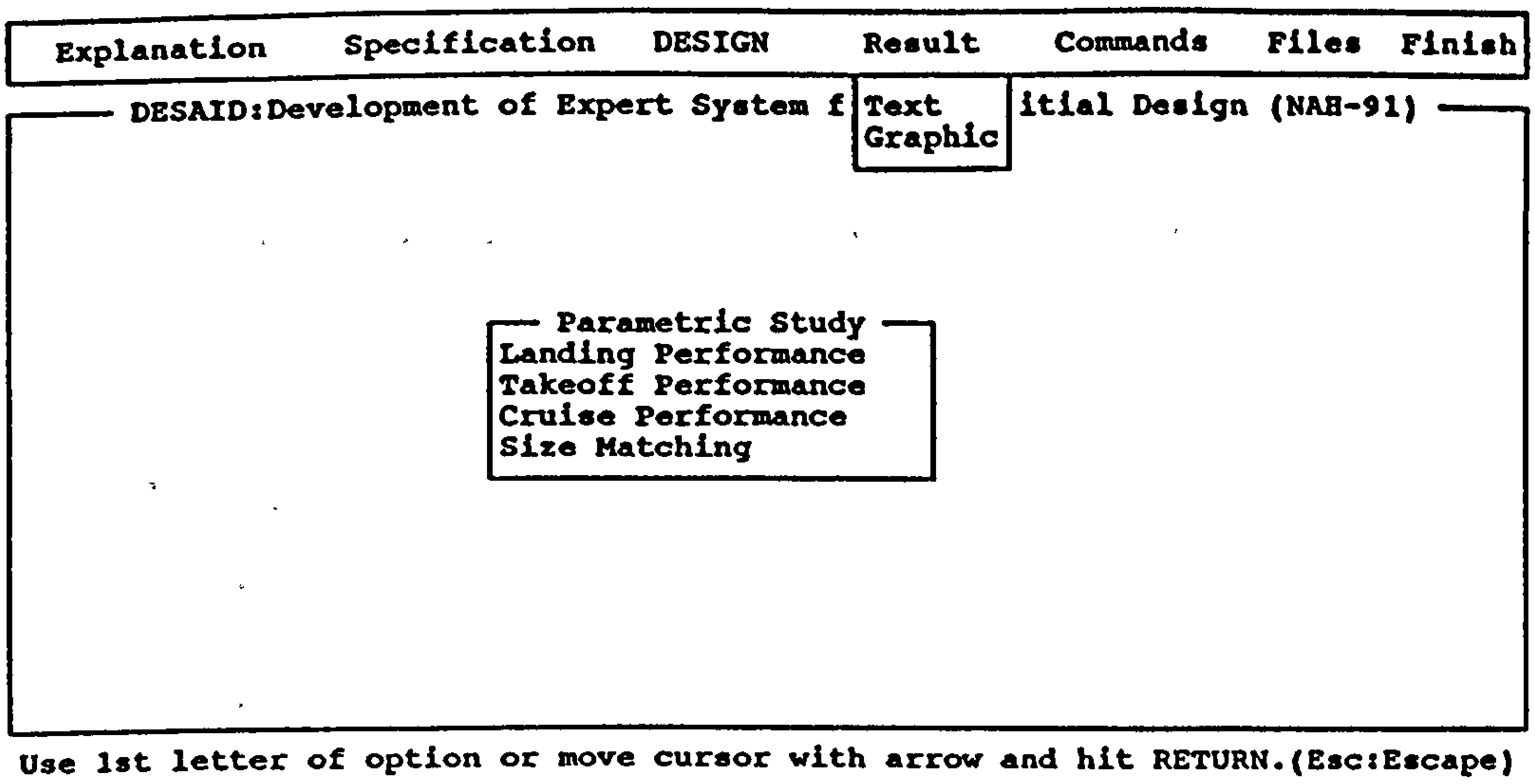
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)

If you select "Graphic", all the output implemented before will be shown for your selection.



Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)

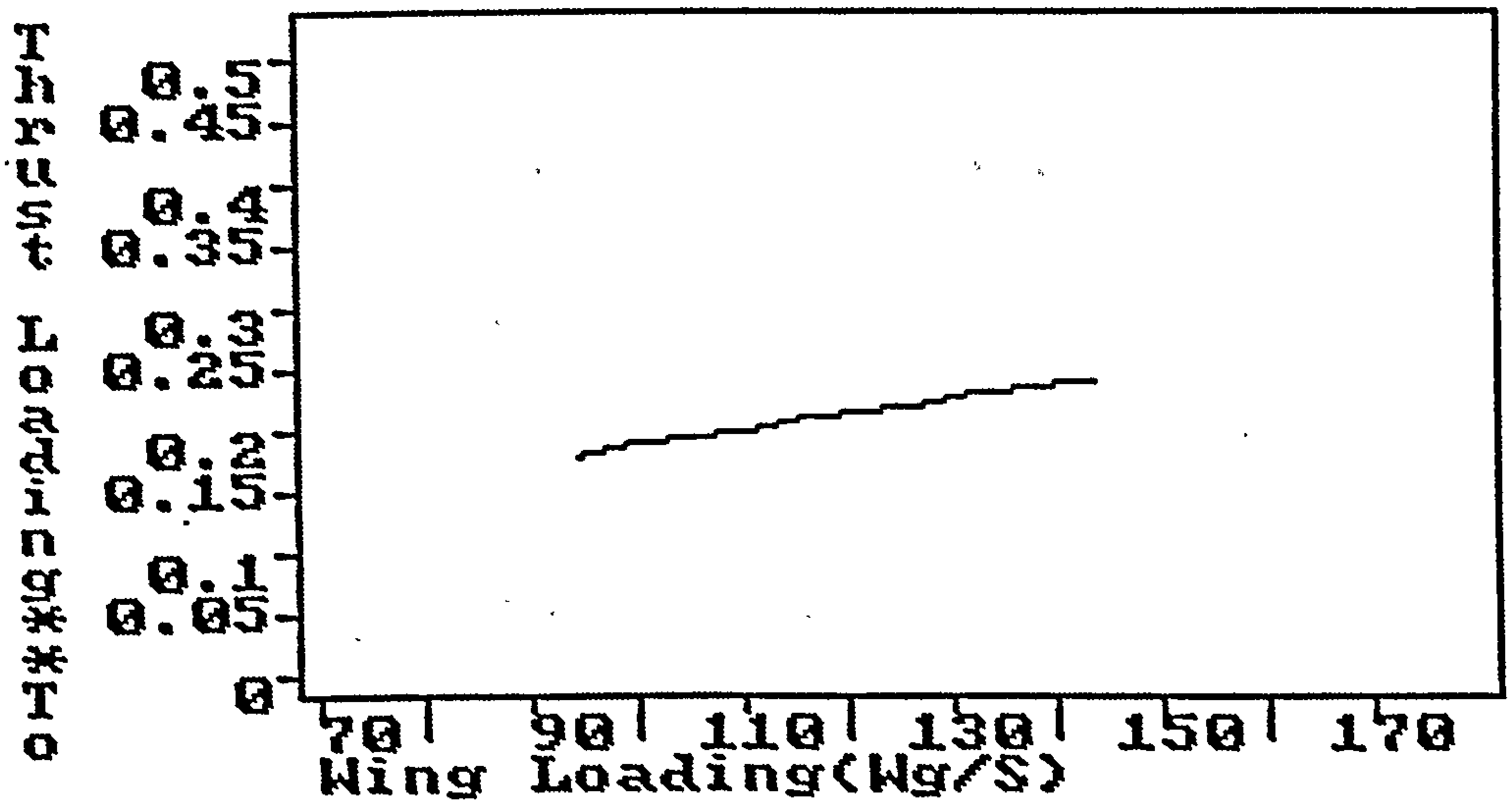
19. If you select the "Parametric Study" and its subsequent nodes as you want, the system will show the graphic results in the order of Landing Performance, Take_off Performance, Cruise Performance, and Size Matching.



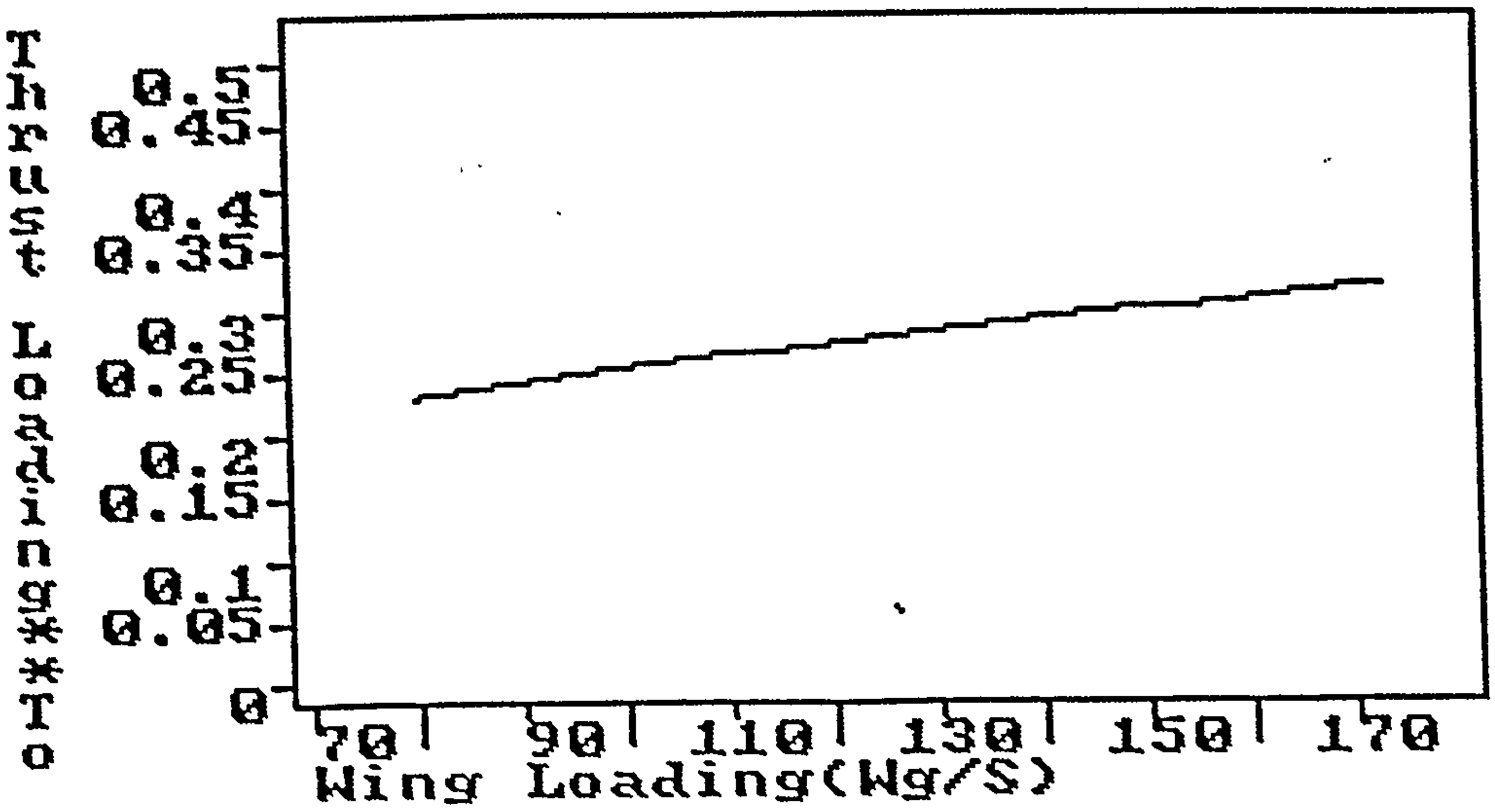
DESAID:Development of Expert System f itial Design (NAH-91)

Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)

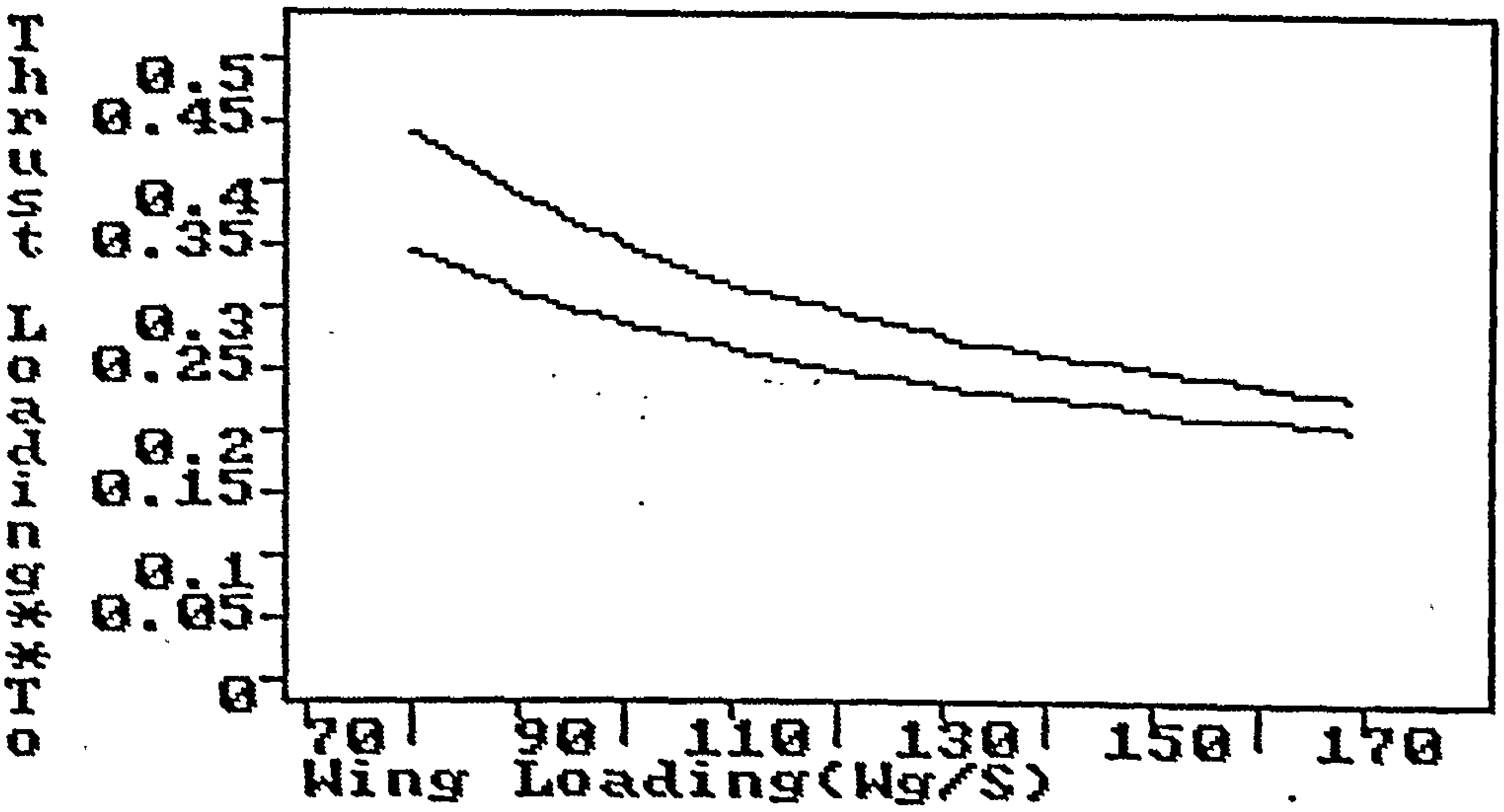
1. The wing loading and thrust loadings of Landing Performance



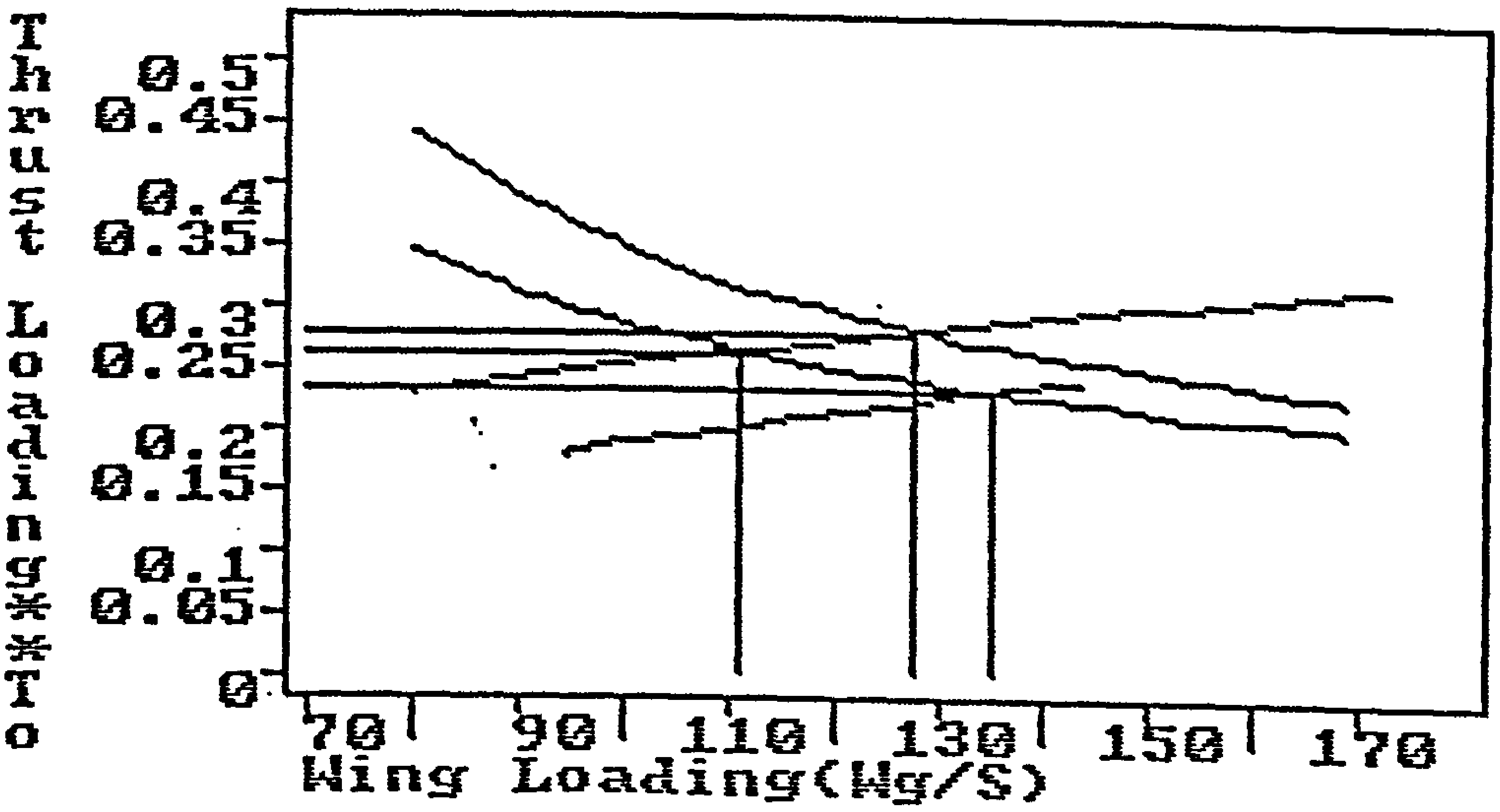
2. The wing loading and thrust loadings of Take-Off Performance



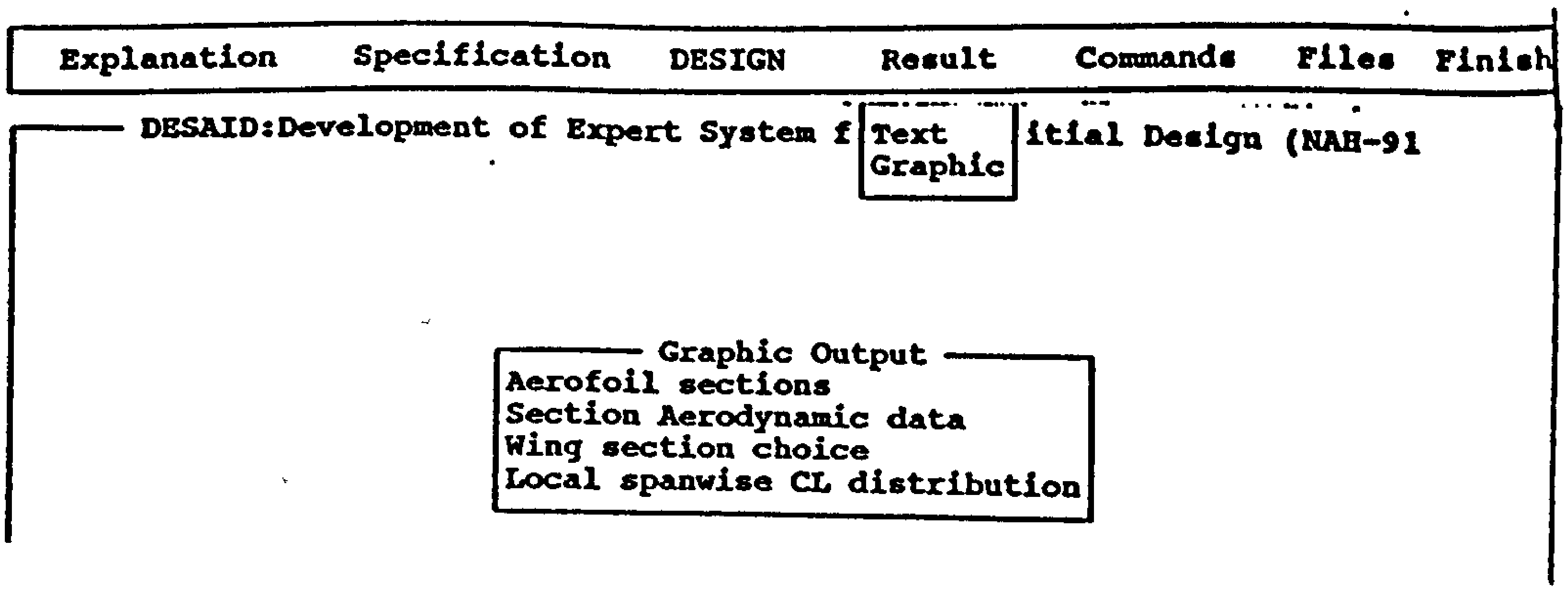
3. **The wing loading and thrust loadings of Cruise Performance**



4. **The wing loading and thrust loadings of Size Matching**



20. In the wing design, the graphic result can be shown in the order of Airfoil series, aerodynamic characteristics, and the result for airfoil selection.

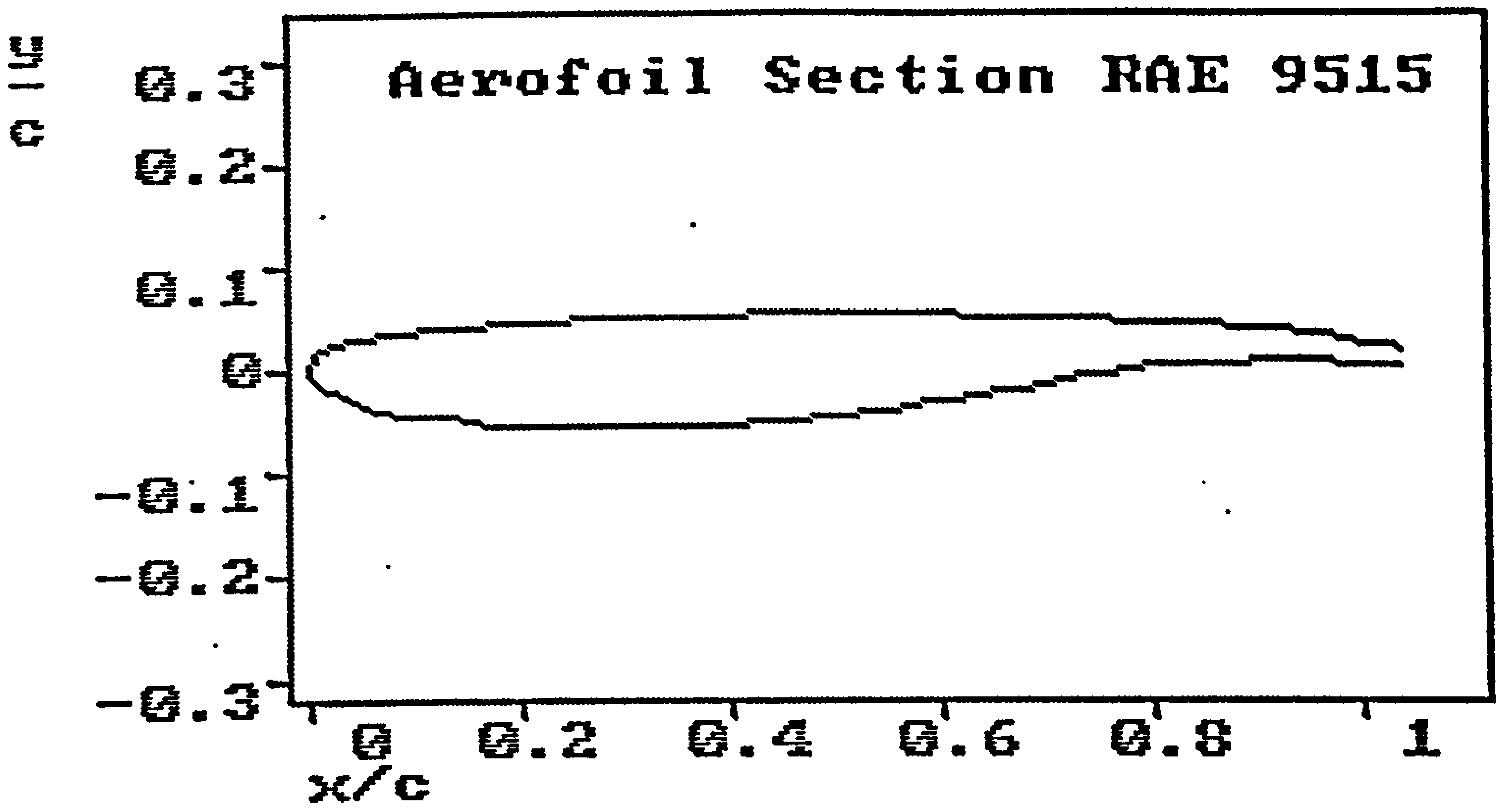


Explanation Specification DESIGN Result Commands Files Finish

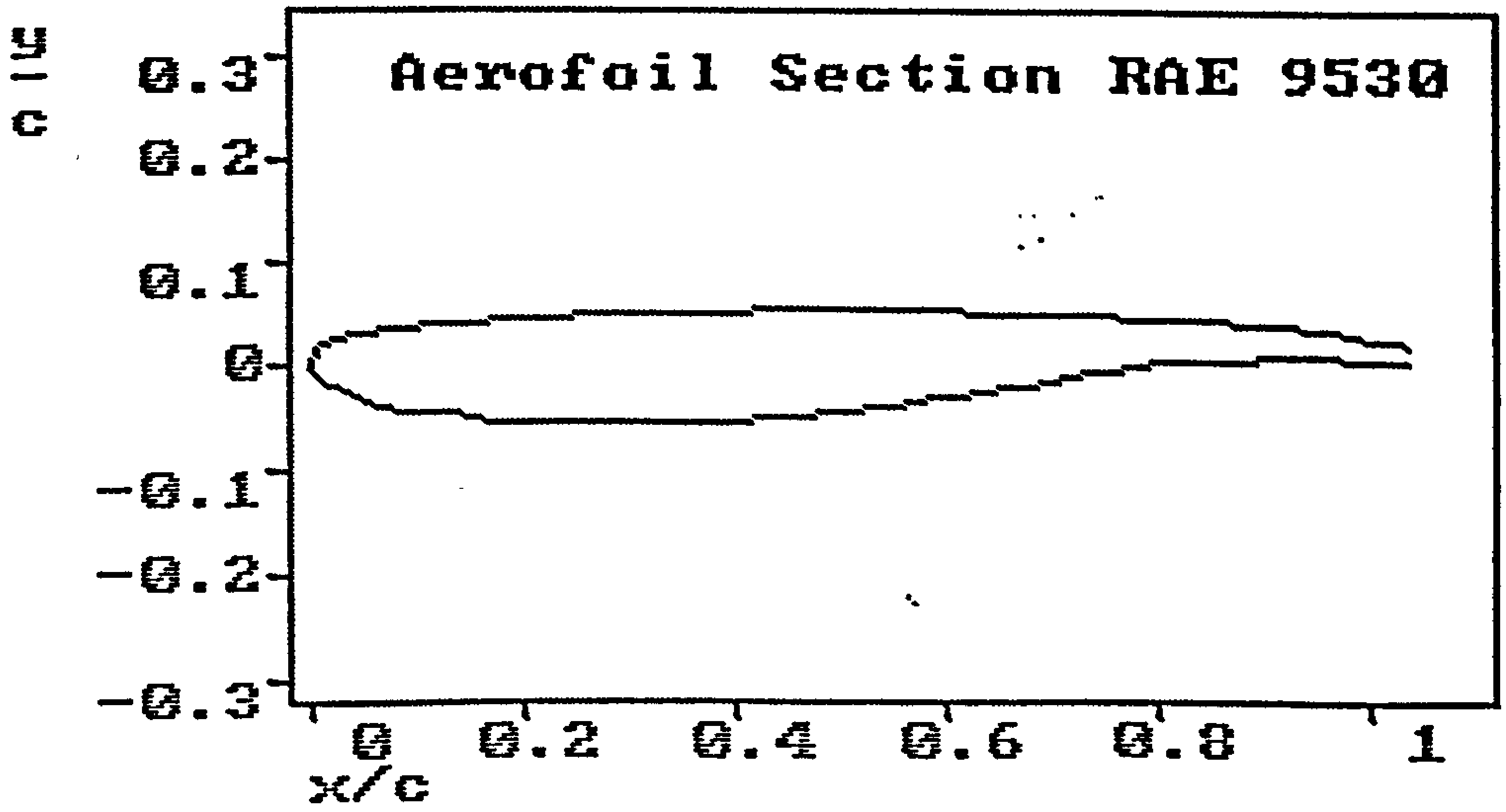
DESAID:Development of Expert System f Text Graphic itial Design (NAH-91

Graphic Output
Aerofoil sections
Section Aerodynamic data
Wing section choice
Local spanwise CL distribution

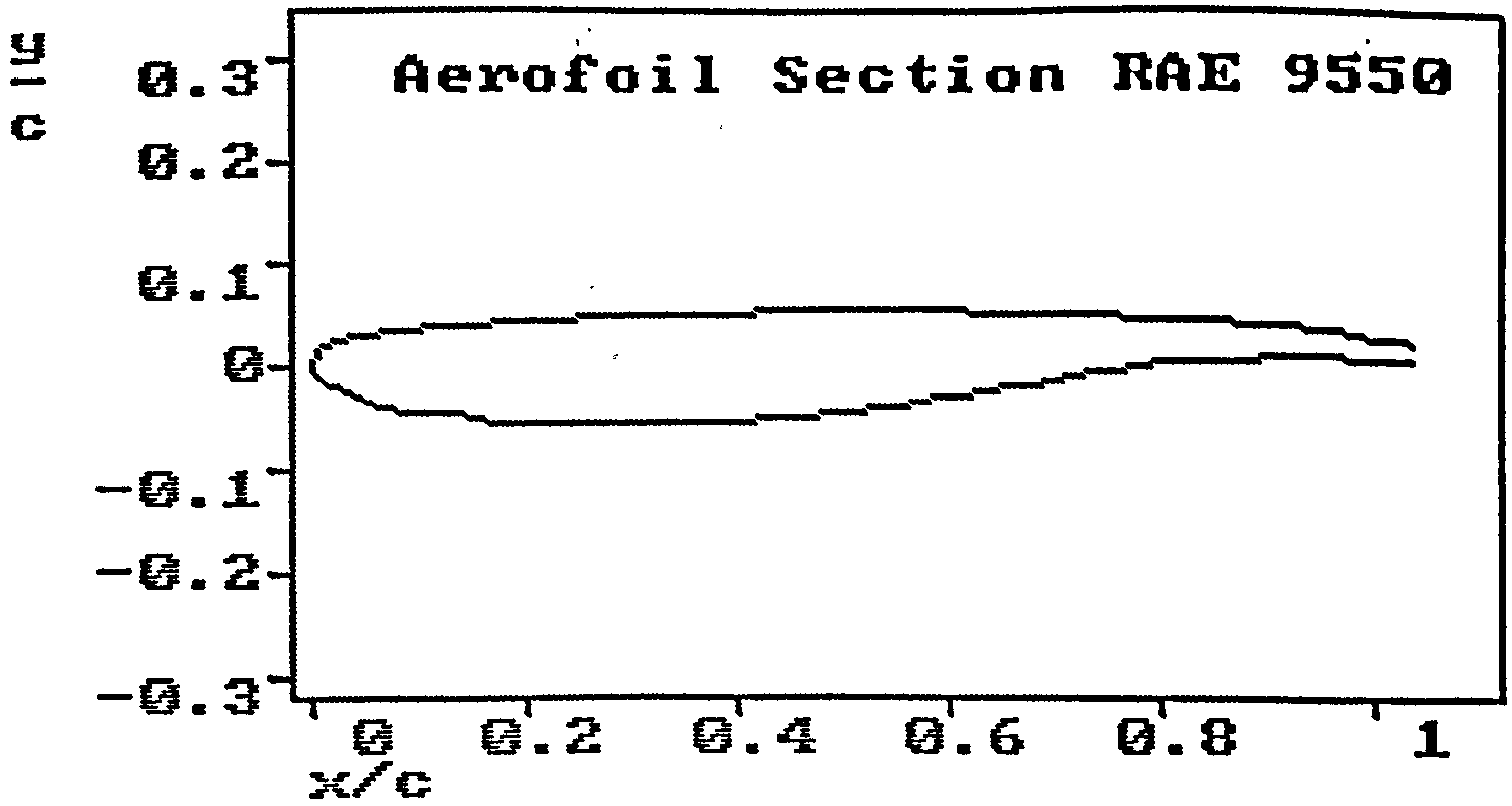
1. Aerofoil RAE 9515



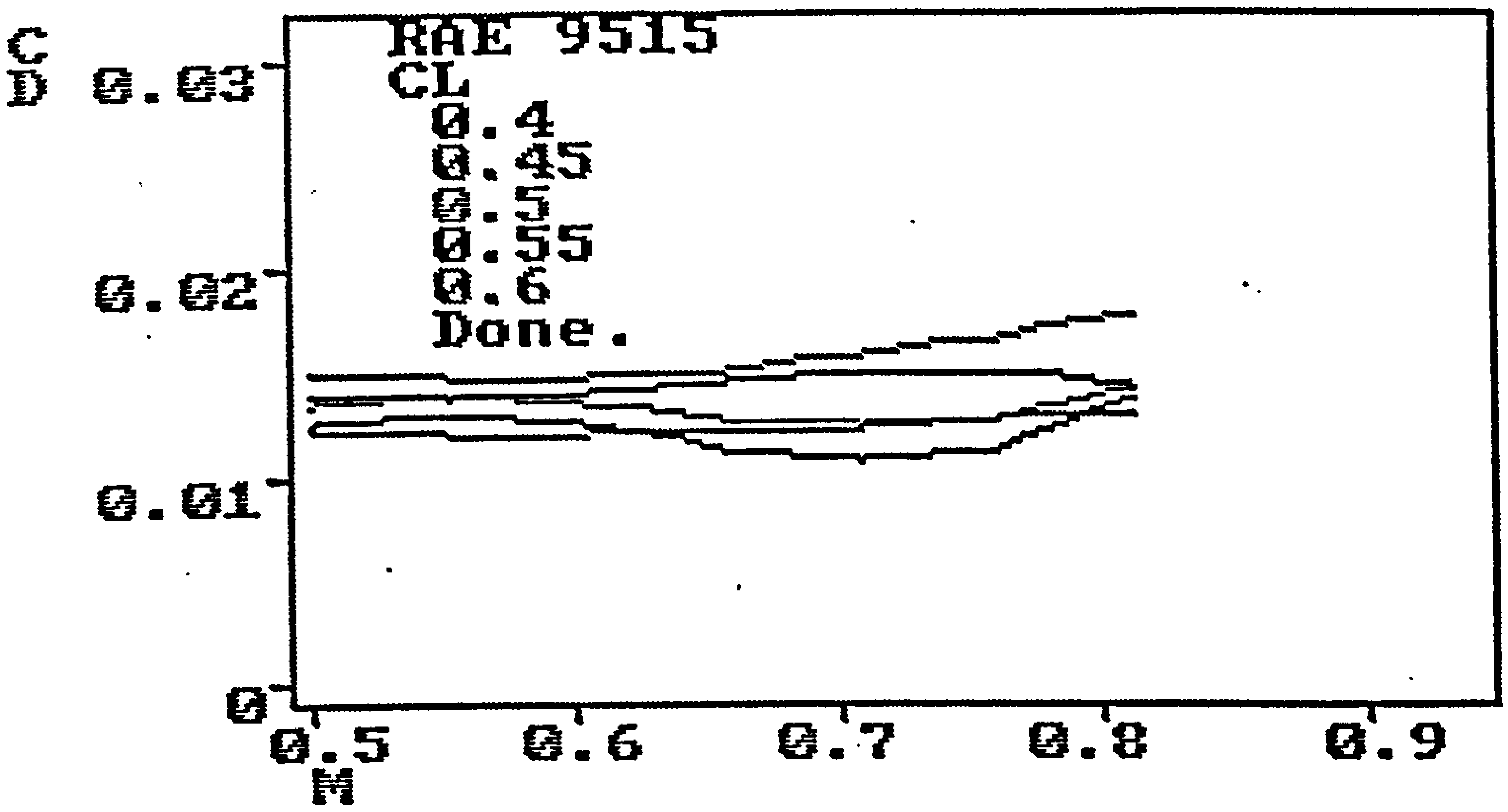
2. Aerofoil RAE 9530



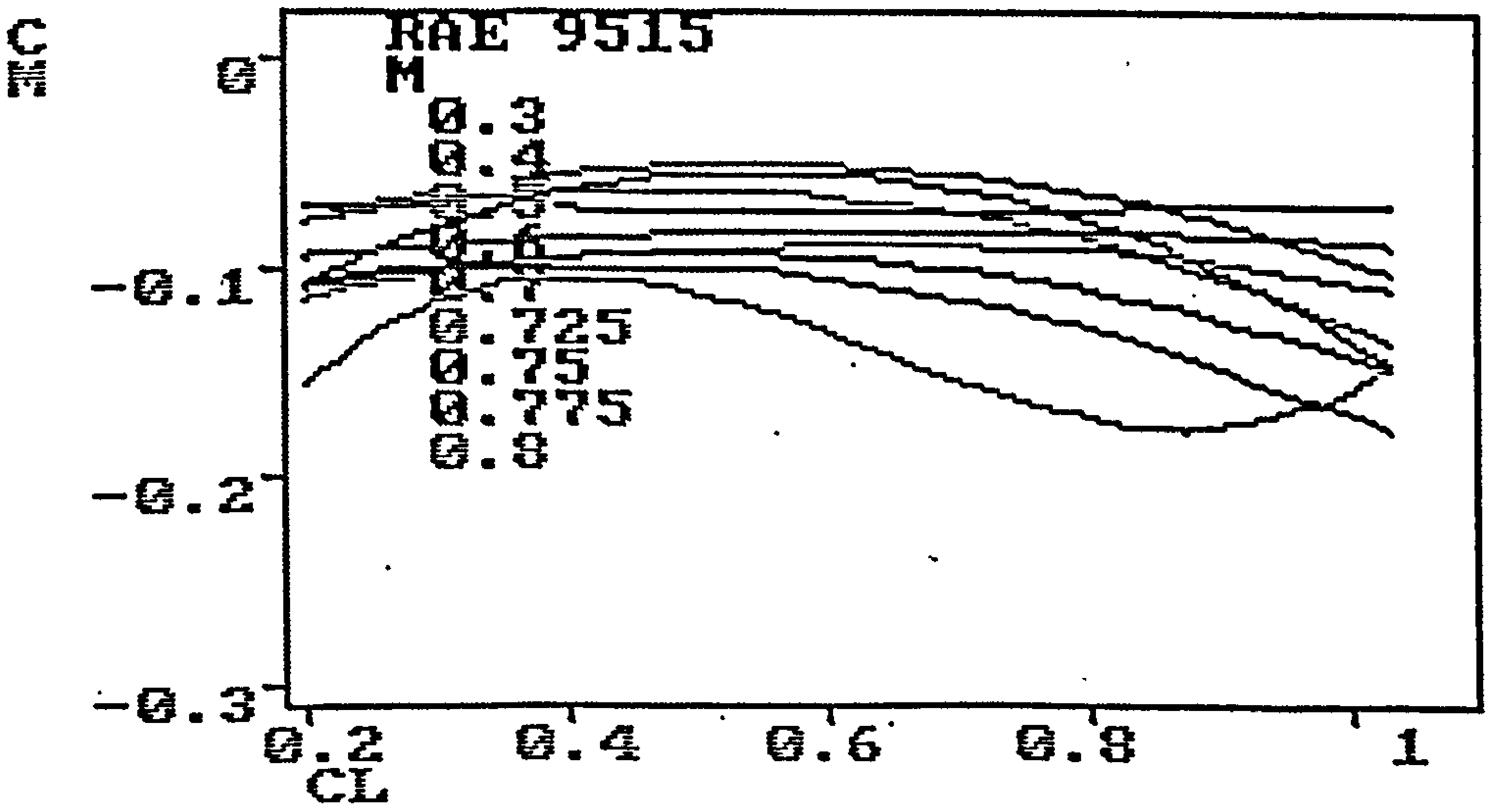
3. Aerofoil RAE 9550



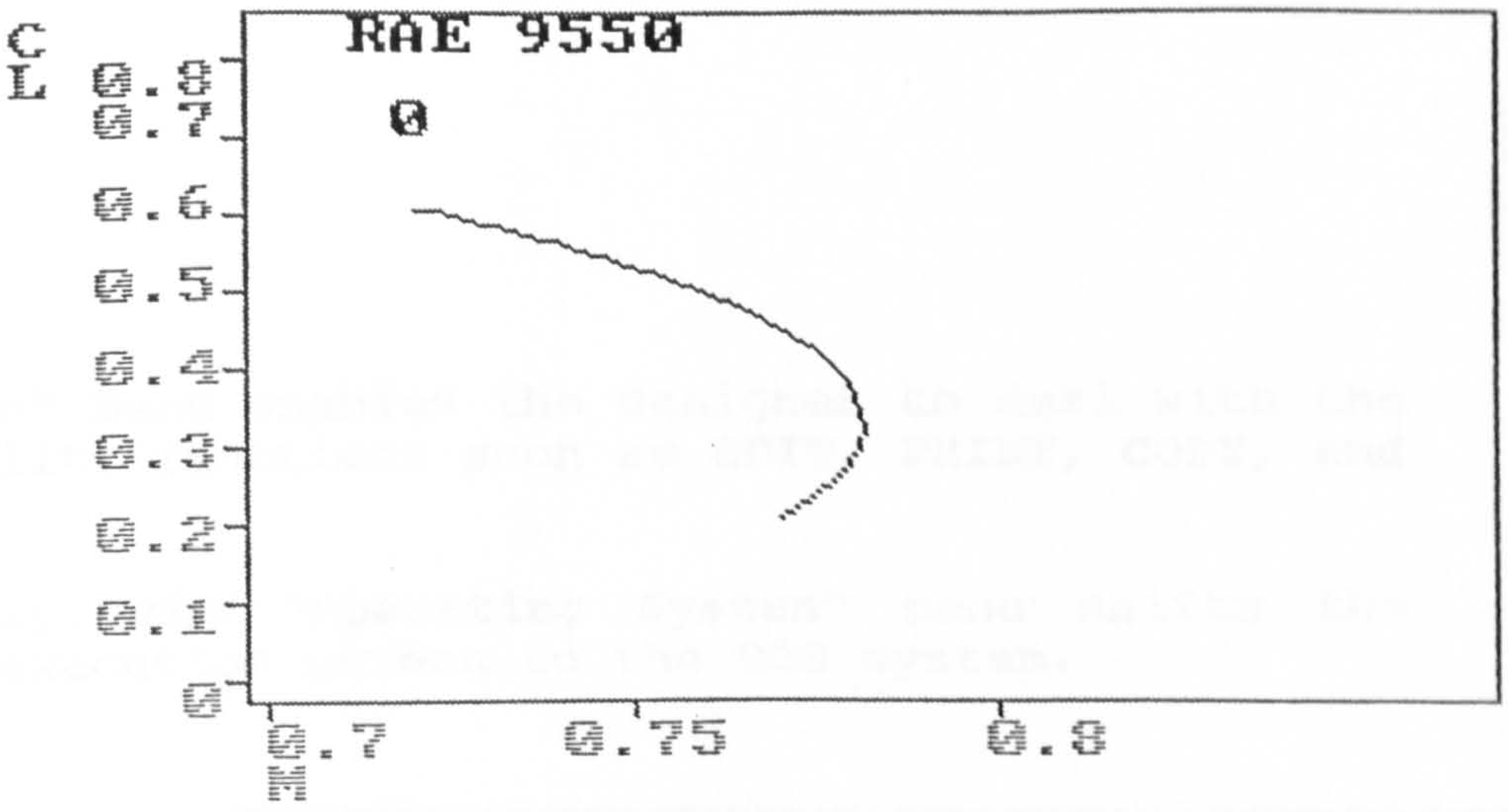
4. Aerodynamic Characteristics (Mach Number vs Drag Coefficient) of RAE 9515



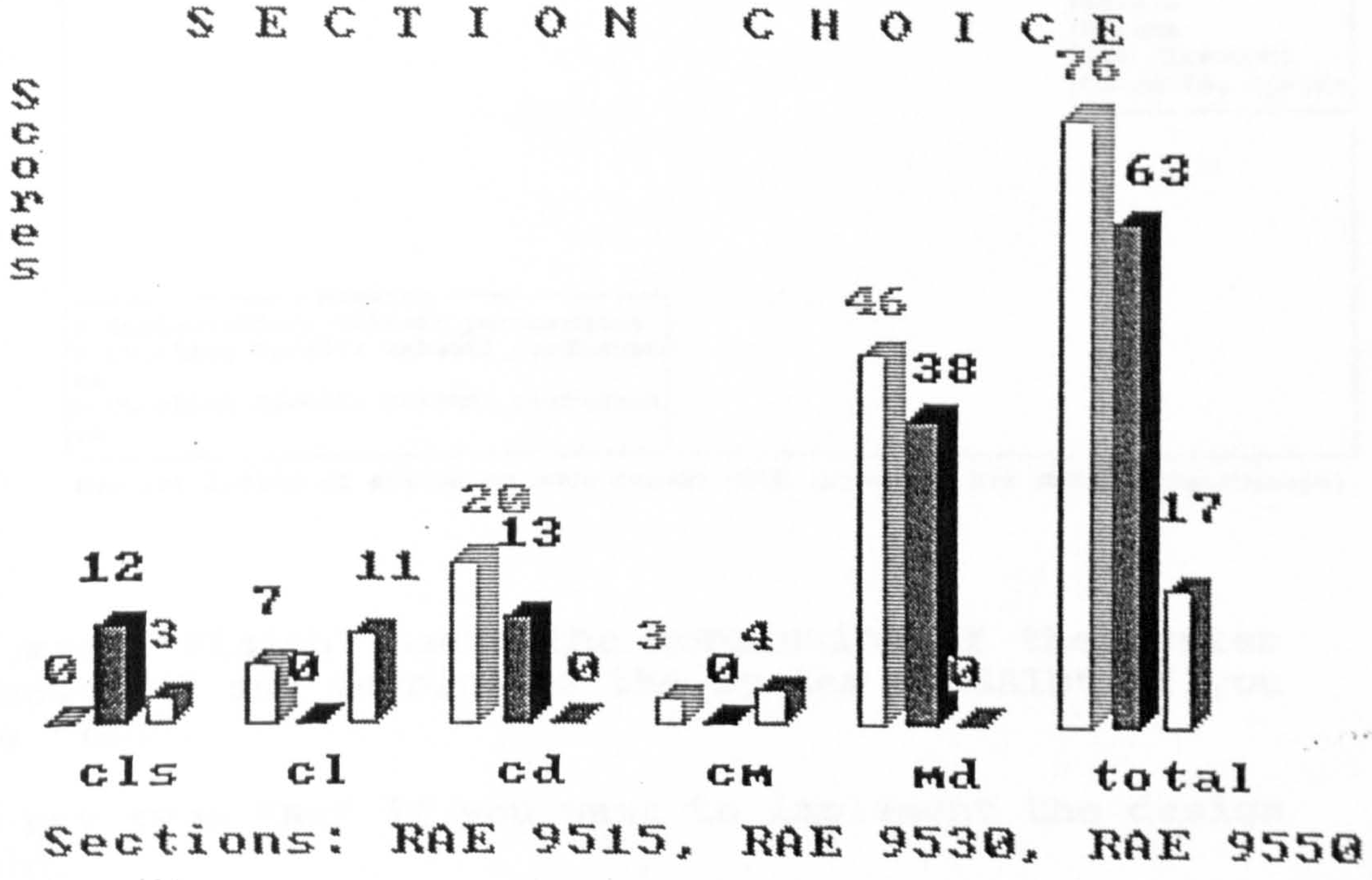
5. Aerodynamic Characteristics



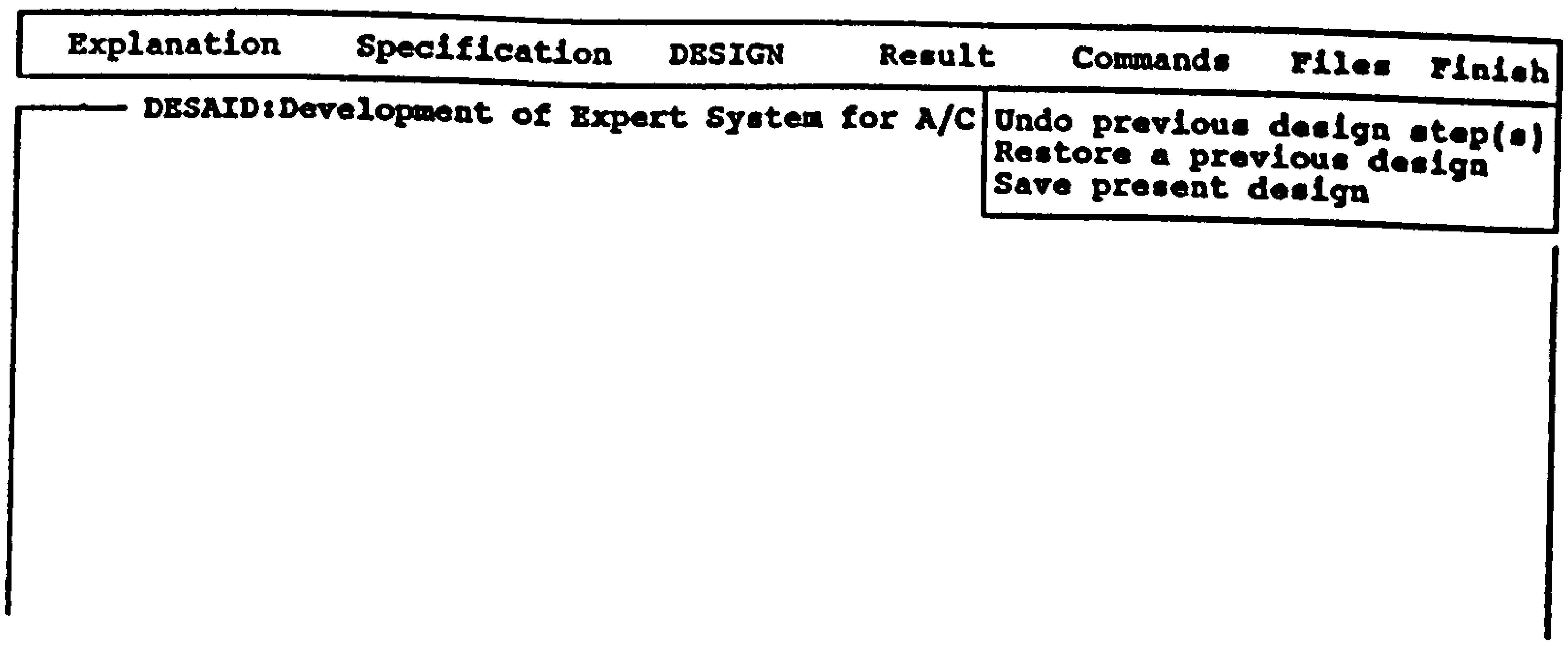
6. **Aerodynamic Characteristics (Mach Number vs Lift Coefficient) of RAE 9550**



7. **The Result for Aerofoil Selection.**

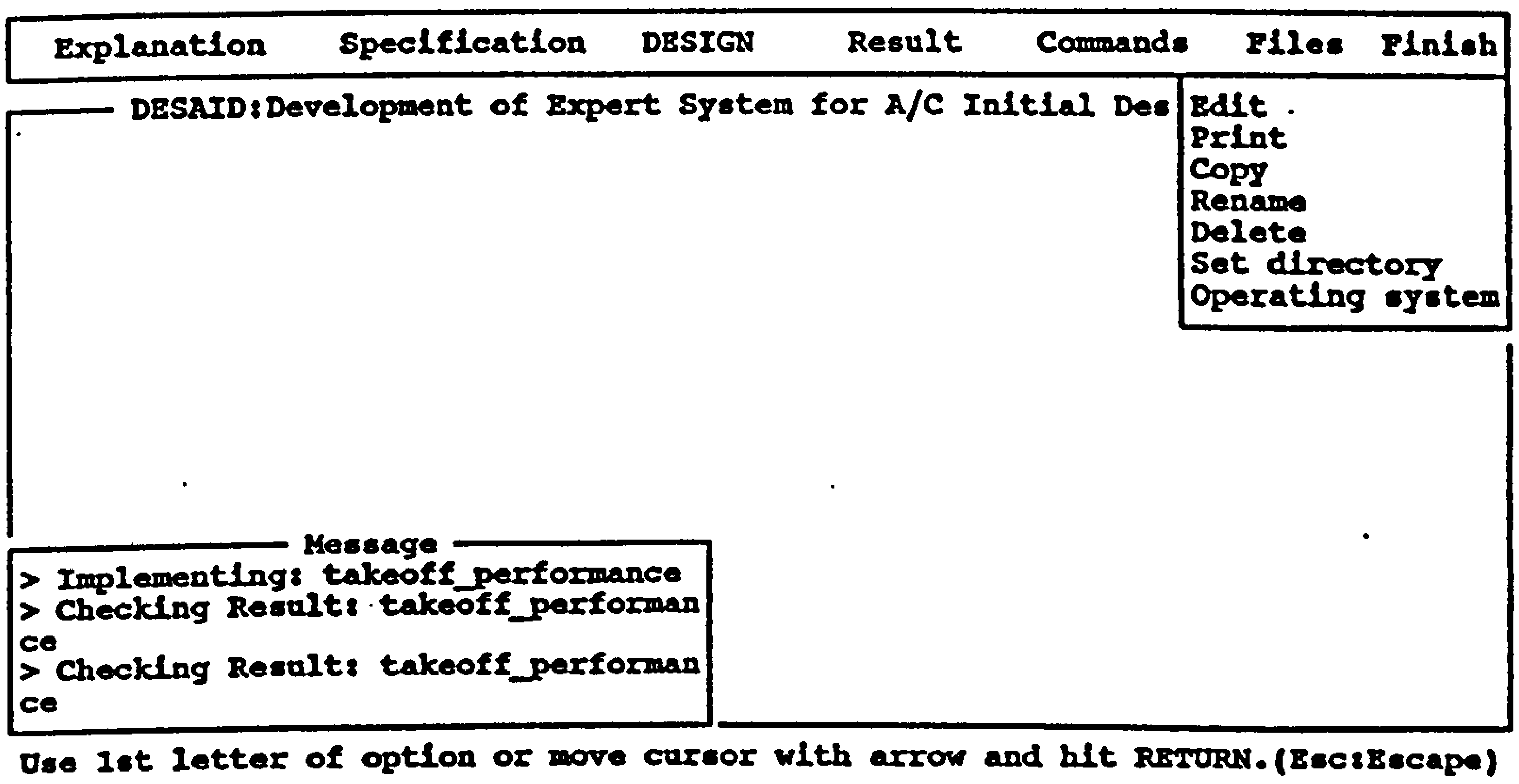


21. In the "Command" pulldown menu, you can select the layer and the node which you want to **re-evaluate and** to save the results.



22. The "File" menu enables the designer to deal with the file utility functions such as EDIT, PRINT, COPY, and so on.

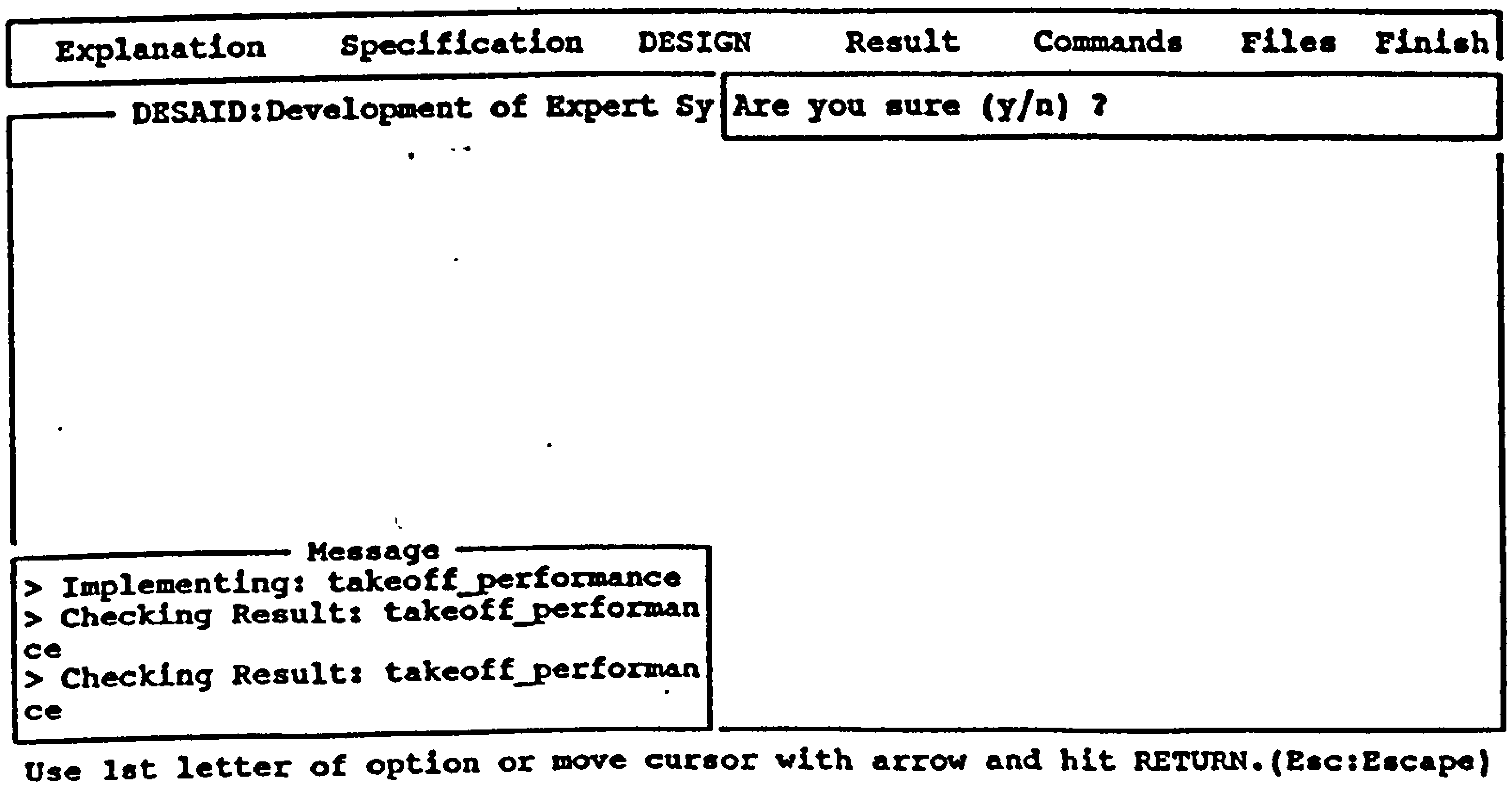
Especially, the "Operating System" menu shifts the current execution screen to the DOS system.



```
> Implementing: takeoff_performance
> Checking Result: takeoff_performan
ce
> Checking Result: takeoff_performan
ce
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```

23. The menu "Finish" means the completion of the system "execution" and terminates the system "DESAID" if you type "Yes".

You may type "No" if you want to implement the design again.



```
Are you sure (y/n) ?
> Implementing: takeoff_performance
> Checking Result: takeoff_performan
ce
> Checking Result: takeoff_performan
ce
Use 1st letter of option or move cursor with arrow and hit RETURN.(Esc:Escape)
```