APPLICATIONS OF COMPUTER–AIDED ENGINEERING TO SUBSONIC AIRCRAFT DESIGN IN A UNIVERSITY ENVIRONMENT

ICAS-86-3.1.1

C. Bil

Dept. of Aerospace Engineering, Delft University of Technology
The Netherlands

ABSTRACT

The Aircraft Design and Analysis System (ADAS), a pilot system for computer-aided conceptual aircraft design, developed at the Delft University of Technology (DUT), is introduced. It is a university contribution to the evolution in aircraft design computerization. Application of design systems for configuration development improves the efficiency and effectiveness of the design process and potentially results in an improved design quality. The ADAS system is based on the philosophy that a CAE system should incorporate the advantages of computer-assistance in design, while retaining the flexibility of traditional design. This paper presents a general overview of the ADAS system and highlights specific components and features, illustrated with examples.

1. INTRODUCTION

During the last two decades the scope and sophistication of computer applications in aircraft design have grown significantly. Recent developments are directed towards integrating disciplinary programs into systems for aircraft synthesis design. It is the application of computer-assistance in all areas of design, from the most inceptual stage, through detailed design, into manufacturing, that represents the Computer-Aided Engineering (CAE) environment.

Fig. 1 gives a classification of current CAE tools according to their dependency on discipline/subsystem, application and design phase[1].

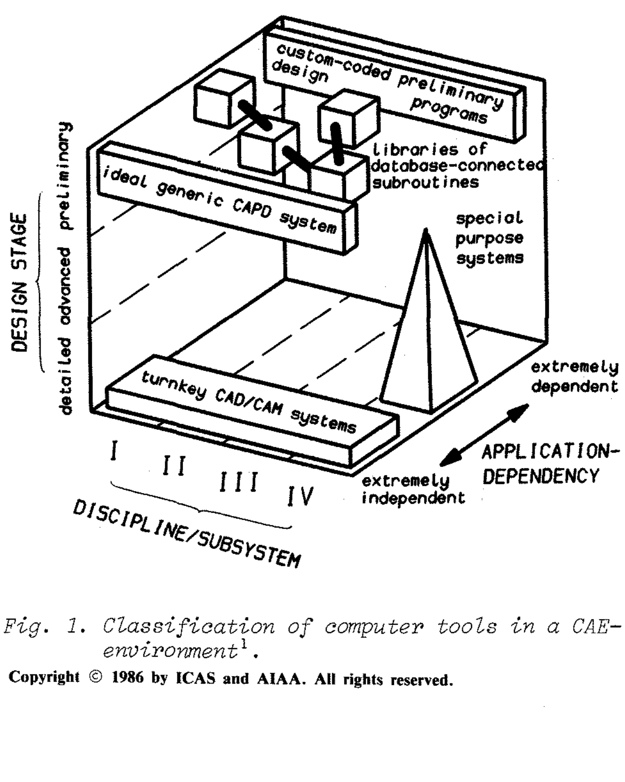


Fig. 1. Classification of computer tools in a CAE-environment[1].

At the far end of the discipline dependency scale, close to manufacturing, are the commercial "turnkey" CAD/CAM systems. In conceptual design there are generally custom-coded (batch) programs.

The incentive to integrate disciplinary programs in a computer-based design system is twofold:

- Improved efficiency and effectiveness of the design process, e.g.:
  - It stimulates consistent use of (standard) design programs;
  - Better development, testing and management of software;
  - Manual data transfer and formatting is eliminated;
  - Automation of routine tasks and processes;
  - More design variations can be evaluated within a given amount of time;
  - Central design data storage in a database implies exact and consistent definition of design parameters;
  - Design data is better available and accessible;
  - Better data protection and management;
  - Immediate insight into current design status.
- Potentially leads to an improved design quality. Design integration presents opportunities for a more systematic approach to design optimization, e.g. (multivariate) implicit optimization.

Penetration of CAE techniques in the early phases of design is hindered by some general factors:

- Problems in the realization of the specific requirements of conceptual design in a computerized design system, e.g.:
  - The heuristic and intuitive nature makes it less suitable for the formalized structure required by the computer;
  - From a management point of view, computer programs may appear less transparent. They may contain hidden or obscured design decisions;
  - The close interaction between designer and computer requires attention to a user-friendly communication interface and ease of operation.
- Financial and organizational consequences, e.g.:
  - Computer hard- and software is costly;
  - Experienced design engineers temporarily have to defer their daily work for training to operate CAE systems;
  - Computer specialists have to be recruited to supervise and maintain hard- and software;
  - Introduction of CAE will undoubtedly have an effect on the companies organization and infrastructure.
- No conclusive proof has yet been given that the use of a design system will result in an improved design. This is primarily due to the inability of current conceptual, mostly statistical, design

methods to accurately and reliably predict physical reality outside their relatively narrow area of application.

The increasing complexity of aircraft projects and the potential economic benefits have induced the aerospace industry to invest in the development of conceptual and preliminary design systems. Thus, design engineers will inevitably become more and more involved in CAE. It is the responsibility of aeronautical faculties to acquaint students with elementary CAE techniques and prepare future design engineers for this new environment. In 1982, 4 engineering departments and the computing center jointly drew up a specification for a new CAD-system, dedicated to CAD research at the DUT. The configuration selected was a multi-user, "turnkey" system, with a PRIME 750 as host computer and running the MEDUSA (CIS) drafting and modelling package. The system, which is designated as the Interfaculty CAD Installation (ICI), became operational in September 1983. A schematic diagram of the ICI hardware configuration is given in Fig. 2. It shows resemblance with the configuration in use at NASA Langley for computer-aided research[2].

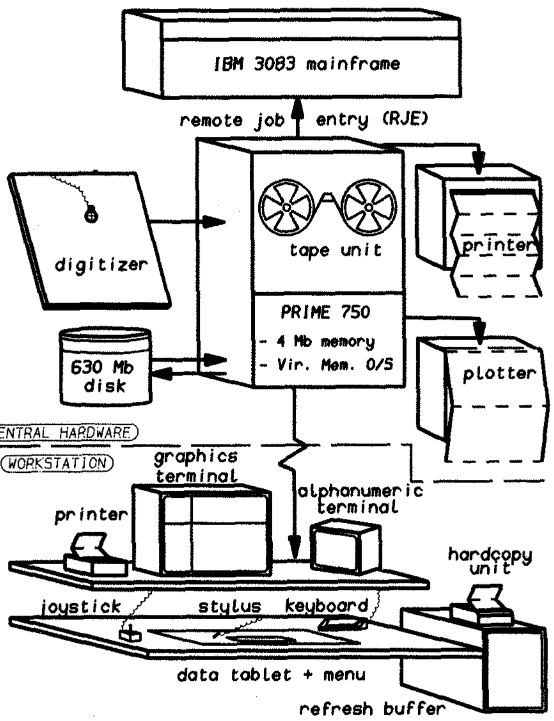


Fig. 2. Delft CAD-system hardware configuration.

Concurrent with the acquisition of the ICI, a research project was initiated at the Department of Aerospace Engineering, to develop specific software modules for conceptual aircraft design. The collection of these programs is referred to as the Aircraft Design and Analysis System (ADAS).

2. AIRCRAFT DESIGN & ANALYSIS SYSTEM (ADAS)

To develop an effective conceptual design system, it is essential to identify and analyze the basic activities that occur during the conceptual design phase. In strict mathematical terms, conceptual design can be simply considered as manipulation of design data, with 4 basic ingredients: independent variables which define a design configuration (descriptive model), dependent variables which represent the design characteristics, design functions (predictive model) which establish relationships between these two and there are criteria to evaluate the design. The primary objective in conceptual design is to assess a global definition of a design configuration that meets all design requirements (feasible design) and, ideally, is optimum according to a given criterion or figure of merit.

The search strategy is basically one of trial and error (iterative). After each cycle, the design is analyzed and evaluated. If deemed necessary, the design configuration and/or requirements are changed and the process is repeated (Fig. 3).

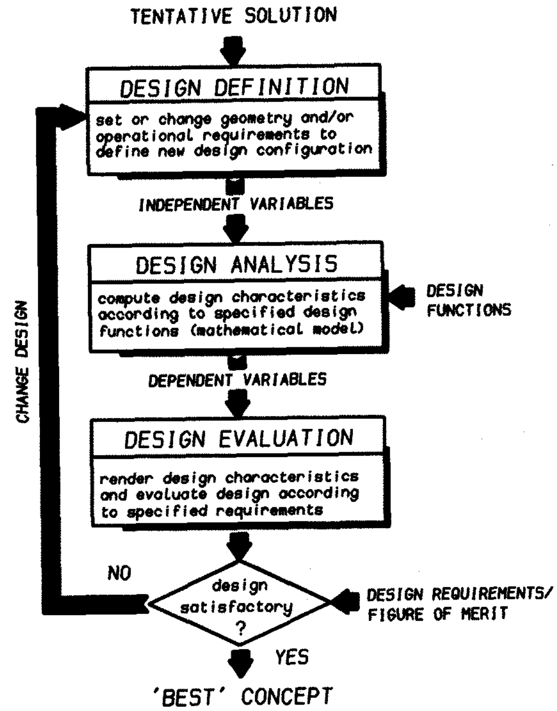


Fig. 3. Basic steps in a design iteration.

There are 3 basic procedures the designer can utilize to find an optimum design[3,4]:

- In the traditional intuitive (optimum) design approach, the designer relies mainly on intuition and experience to select and change design parameters. The principal advantages (+) and disadvantages (-) of this approach are:

  + The designer can make full use of experience, augmented by proven and simple design methods;
  + Simple or no programming is required;
  + Number of designs to be analyzed is limited;
  + Maximum use is made of calculated results;
  + No a priori choice of one merit function;
  + Arbitrary, though limited number of variations and design modifications;
  - No guarantee that a real optimum is obtained;
  - No useful result outside the designer's experience;
  - It is time consuming: the designer will therefore tend to resist desirable changes in the design specifications or other previous decisions;

- With explicit or parametric optimization, a multitude of designs, each with different parameter values, are generated and analyzed. All designs are subsequently evaluated and the "best" design is selected. The advantages (+) and disadvantages (-) are:

  + It is rooted in the industrial approach;
  + Requires relatively simple programming;
  + The designer has complete control over decisions;
  + No a priori choice of a single merit function;
  + Sensitivity of off-optimum design conditions remains visible;
  - No guarantee that a global optimum is obtained;
  - Only practical for a limited number of independent variables, 3 or 4;
  - Many designs are evaluated, only a few are actually used;
  - The designer is not encouraged to extend the number of variables;
  - Changes in the design specification make previously generated results obsolete and are resisted;

- Implicit or multivariate optimization requires the design process to be fully automated. The figure of merit and design requirements are quantitatively formulated as an objective function and constraints respectively. An optimization algorithm (optimizer) changes the specified design parameters (free variables) based on mathematical information acquired during the optimization process. The advantages (+) and disadvantages (-) are:

  + It potentially leads to an improved design quality due to the rigorous approach;
  + Especially useful for multi-variable systems;
  + Effect of "biased" decisions is eliminated;
  + Useful for unconventional configurations and program structures;
  + Changes in design specifications are easily met;
  - Programming and debugging are difficult;
  - Optimization algorithms are not always effective;
  - Convergence problems may occur: no solution is found;
  - No insight into design sensitivity is obtained;
  - Inexperienced designers may produce and accept unrealistic results.

By comparing the pros and cons, it can be concluded that these 3 optimization techniques are complementary, hence, an effective design system should give the designer the freedom to choose a suitable combination for a given design problem. The following general system requirements were drawn up for ADAS:

- ADAS must be capable of accommodating many user-defined problem structures and the procedures to solve them (flexibility);
- Design functions must not contain design decisions (transparency);
- Attention should be given to a user-friendly communication interface and ease of operation;
- ADAS must be open-ended to interface programs and to proceed to downstream design levels;
- Parameter sensitivity and multivariate optimization studies should be optional;
- Simple and rapid data pre- and postprocessing;
- ADAS should be implemented on the ICI and preferably make use of available software (graphical and numerical subroutine libraries).

A schematic overview of the general ADAS architecture, as developed on the basis of these requirements, is given in Fig. 4.

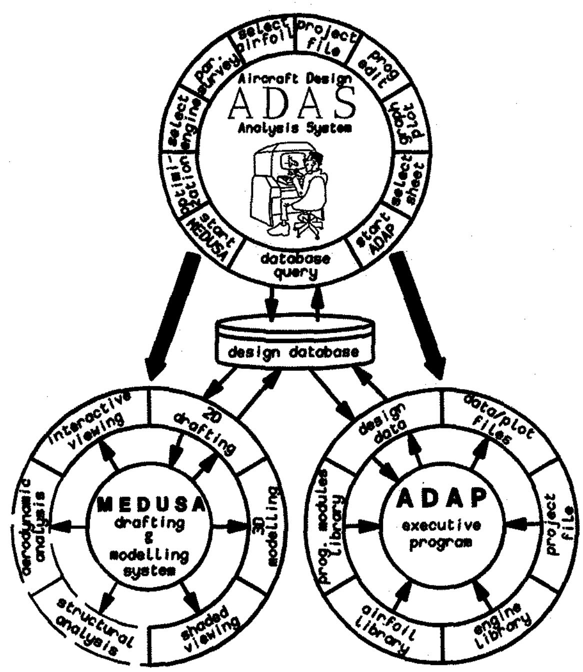


Fig. 4. General ADAS system architecture.

The system comprises a trinity of programs:

- ADAS controls the user-system dialogue. All processes and subsystems can be invoked from ADAS command level. Activities under direct control of this program can usually be classified as preparation for analysis or postprocessing of analysis results, e.g. database query, graph plotting, etc.;

- ADAP is a general-purpose executive program, which controls the execution of user-supplied analysis programs. Optionally, ADAP can be run in parametric survey and/or optimization mode;

- MEDUSA is a general-purpose drafting and (solid) modelling system. The ADAS system features a 2-way interface to exchange design geometry information between a MEDUSA drawing database and ADAP.

In the following sections, these 3 system components will be discussed in more detail. To illustrate the functional relationships and to highlight specific features, the ADAS system will be presented in this paper as proceeding through the stages of a typical design process.

3. ADAS: DESIGN DEFINITION

The ADAS program is the system kernel. It controls user-system dialogue through a command-oriented language. Available commands are operating system commands, which are used typically for file editing and management, and specific ADAS commands[5].

Users can select the keyboard or command menu tablet interchangeably as input device. Each user owns a command file, which contains information regarding ADAS commands, which the user is authorized to use and a description of the command menu layout and its field definitions. ADAS allows the user to rearrange the command menu or rename commands and so create a personalized command environment.

3.1. MEDUSA: design geometry definition

The MEDUSA drafting and modelling system is a collection of program modules configured around a 2D drafting module. At drafting command level, an existing drawing can be retrieved for modification (drawing editing) or a new drawing can be created. MEDUSA features numerous options to perform basic and complicated drawing functions, e.g. automatic cross-hatching and dimensioning, different line and text types, transformation and duplication of drawing parts, parametrics, etc. To add information to a drawing, one first selects a graphical element class, i.c. line, text or symbol (prim). Each element class is in turn associated with a set of attributes, e.g. line or text type, number of points in a line, curve fit weight factors, layer number, position coordinates, etc., which define an individual element.

A schematic 3-view configuration drawing can be input using standard MEDUSA features (Fig. 5).

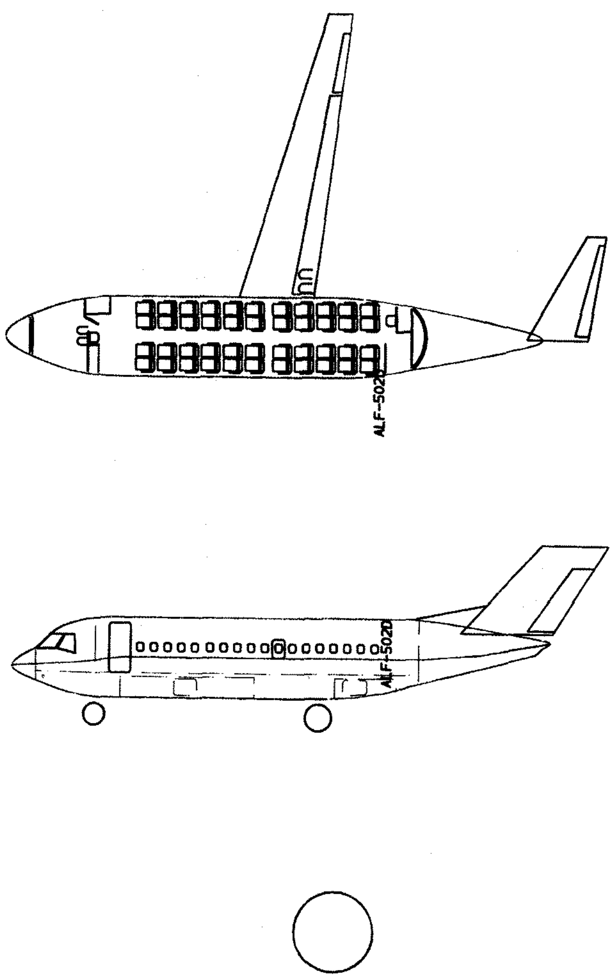


Fig. 5. Schematic design geometry definition.

However, some simple rules have to be observed to make the drawing compatible with the ADAS-MEDUSA interface program, i.c.:

- Each element in the drawing is assigned a predefined layer number, used by the interface program to identify specific aircraft components, e.g. wing, fuselage, tailplanes, etc.;
- Specific drawing conventions are in effect for each component with respect to the number of lines, the number of points in a line, the sequence of points, use of arcs and conics, etc.

After completion of the drawing, it is assigned a sheet name and saved as a drawing database in the User File Directory (UFD).

3.2. The ADAS design database

An ADAS database is a computer-based repository of all engineering information related to one design. Because of the relatively limited amount of data involved in conceptual design, a simple and straightforward direct access method has been adopted for the database structure. This concept may be changed when a general-purpose (relational) Data Base Mangement System (DBMS) is implemented.

Information in a database can be subdivided into several logical data subsets, i.c.:

- Design engineering data in an ADAS database primarily consists of 10000 single-precision real variables, of which only a small subset is currently in use. Individual data entries can be referenced simply by their corresponding index number. One master Data Dictionary File (DDF) has been compiled which forms a cross-reference between the values in a database and their corresponding physical definitions, i.c. a textual description, physical unit and print format (number of decimal digits)[6]. Fig. 6 illustrates the function of the DDF for database query.

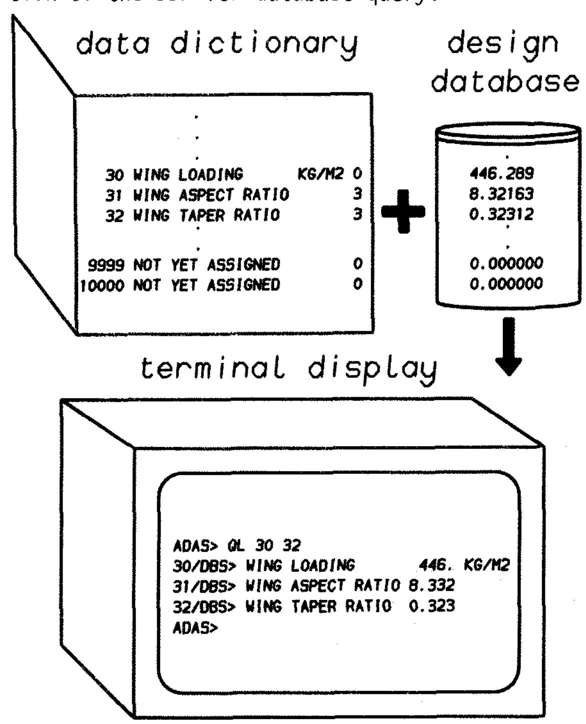


Fig. 6. Dictionary for database data description.

- Design-associated data files can be specified by saving the corresponding file names in the database, i.c.:

  - The MEDUSA drawing database which contains design geometry information (cf. 3.1);
  - An engine file which contains nacelle/engine geometry information and performance data of the selected (reference) engine. Engine files are stored in a separate directory (engine library). Engines can be "rubberized" with simple scaling rules;
  - Upto 10 airfoil files which contain section thickness distribution and aerodynamic data. Airfoil sections can be specified for all lifting surfaces, i.c. wing, tailplanes and/or canard. Airfoil files are stored in a separate directory (airfoil library).

- ADAP control directives have to be specified in the database, according to the required ADAP control mode(s) (cf. 4.1), i.c:

  - In the parametric survey mode (explicit optimization), the analysis program will be repeatedly executed with different values for 1 or 2 user-specified design variables (survey parameters). Survey parameters are selected by specifying the corresponding database index and the variation range limits. The specified design characteristics to be analyzed for each design point are referred to as survey target functions;
  - In the optimization mode (implicit optimization), control is passed to a numerical optimization algorithm for objective function mini- or maximization by variation of free variables and subject to nonlinear equality and/or inequality constraints. Free variables are specified by their corresponding database index and bounds on the search range.

  Control modes can be selected in combination (nesting) or omitted to execute the analysis program only once (design point analysis).

After completion of the input definition, the database is saved and filed in the UFD.

3.3. The analysis program

A computer algorithm for solving a particular design problem, is referred to as an analysis program. In the ADAS system, analysis programs are developed, in FORTRAN 77, and supplied by the user. Admittedly, some experience in FORTRAN programming is required, however, considering the level of computer experience of potential ADAS users and the gain in flexibility in problem definition, this concept seems justified[7].

Standard design and performance computations[8] have been preprogrammed and are available from a program library as subprograms. They can be included in the user's analysis program as disciplinary "building blocks" to compose a simple or complex synthesis program, depending on the design problem at hand. The program library also contains special print and plot routines to output data that is generated, but not saved in a database. Survey target functions, objective function and/or constraints may be any item computed in the analysis program and are not restricted to database entries. An analysis program can e.g. look like this:

```
CALL D$CGFL       /* Compute CG shift due to fuel
CALL D$CGSH       /* Compute CG shift due to pay-
                     load
CALL D$PL09       /* Draw load and balance diagram
CALL D$PIVI       /* Compute pilot's field of vision
CALL D$PL11       /* Draw pilot's field of vision
CALL D$PCOH (0.1,1.0,0.4,0.8,-0.05,0.1,10)
                  /* Set flight conditions for aero
CALL D$POLH       /* Compute high speed aerodynamics
CALL D$PR20       /* Print lift coefficients
CALL D$PR21       /* Pring drag coefficients
CALL D$PRNG (V(474),V(475))
                  /* Compute payload - range
CALL D$PL01       /* Draw payload - range diagram
```

Design knowledge, embedded in prediction methods (design functions), resides outside the ADAS system. Thus, modifying these methods does not affect the generic ADAS structure. Moreover, no restrictions are imposed on detail or sophistication. In fact, several modules may exist to compute one particular item, but with different degrees of accuracy. User program modules can be put into the program library in order to make them available for general use.

Only a few basic conventions, in terms of variable names, use of common blocks, etc., have to be considered to develop and implement a new program module. For example, the engineering data from a design database is stored in a FORTRAN array data structure V(1:10000) in common block /A$ADAS/. Individual elements in the database can be referenced by the corresponding array element V(index), where index is the database index.

3.4. The project file

To execute the ADAP program, it must be provided with file names for data in- and output. These have to be assigned to global system variables which are stored, among other information in a special file, referred to as a project file, e.g.:

variable name	example value	description
.IDBS	FOKKER-100.DBS	database name
.DSPR	WING.DSPR	analysis program
.COMO	WING.COMO	output data file
.PLOT	WING.PLOT	output plot file
.COMF	MYCOMMAND.COM	command file
.NAME	C. BIL	designer name
.PROJ	WING OPTIMIZATION	project id.
.DATE	13:20:44/25 Mar 86	last session

Entries in a project file can be displayed or changed by system variable name. Not only is the project file a means of communication between the ADAS and ADAP programs, it also provides an up-to-date list of all data files associated with a particular design study, hence the name.

4. DESIGN ANALYSIS

Before the user-supplied analysis program can be executed, it must first be compiled and loaded into an executable file, together with the ADAP main program object code and subprograms in several libraries, e.g. the program library. The compile + load step is automated and is only required if a new or modified analysis program is to be processed, in all other cases the ADAP executable existing in the UFD can be run directly. As ADAP is a self-contained program and does not require intermediate user response, it can be processed either interactively, as a background process (phantom) or in batch.

During ADAP execution, data is retrieved from the design database specified in the given project file and subsequently from the design-associated files specified in the database. The input/output sequence is schematically illustrated in Fig. 7.

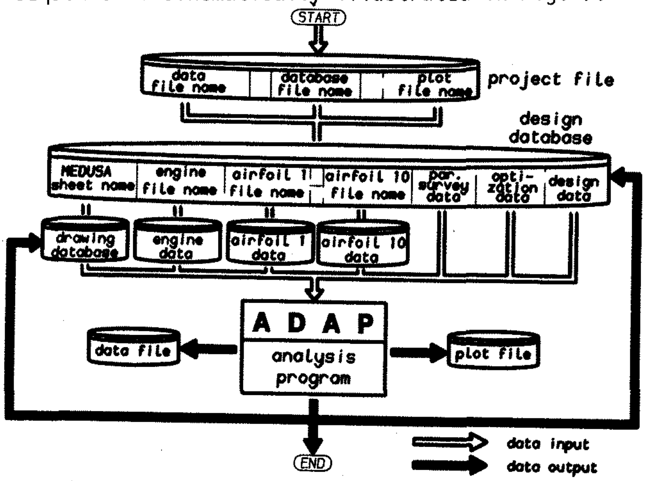


Fig. 7. ADAP data input and output from and to distributed files.

4.1. ADAP control modes

After data input, ADAP starts the actual processing of the user-supplied analysis program. The control flow of this part of the ADAP program is given in Fig. 8

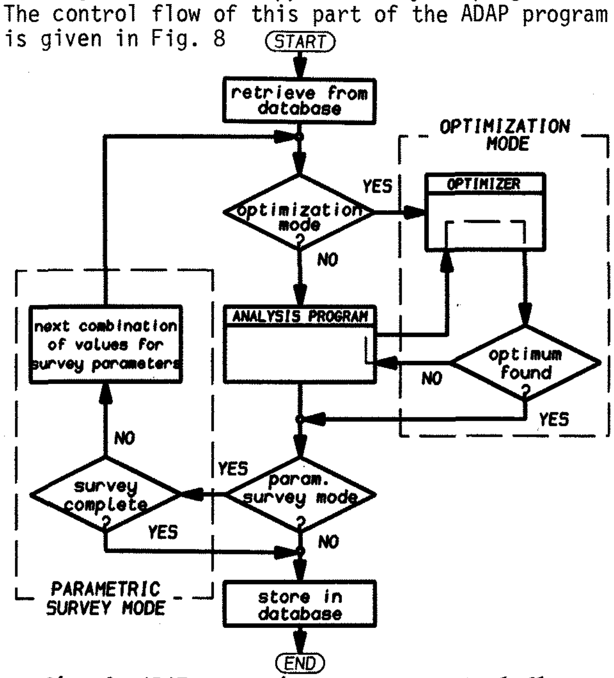


Fig. 8. ADAP executive program control flow.

4.1.1. Design point analysis mode

The most simple use of ADAP is when the analysis program is executed only once, e.g. to analyze a given design configuration. Fig. 9 presents some typical design point analysis results, produced with the program listed in section 3.3.

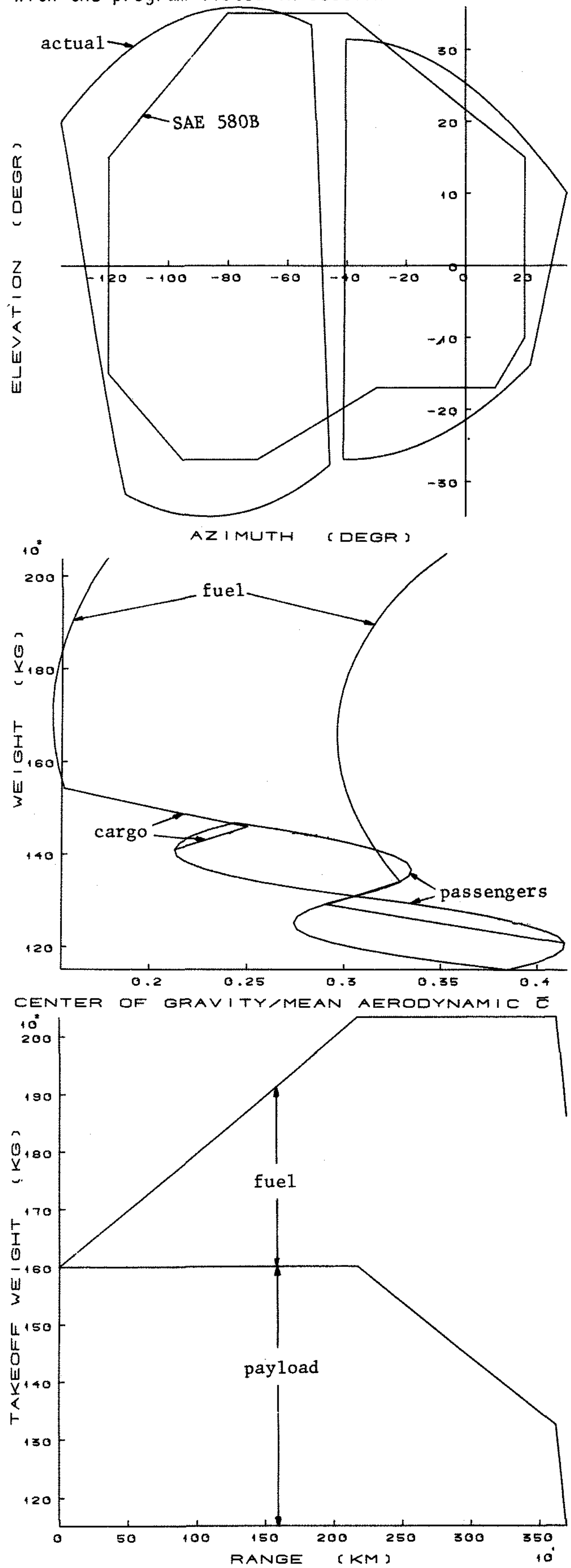


Fig. 9. Design point analysis results.

4.1.2. Parametric survey mode

In parametric survey mode, values are determined for each survey parameter by dividing the specified variation range in upto 10 equidistant points. The analysis program is executed for each combination of values and the survey target functions therein are evaluated and sampled (nested loops). When all surveys are completed, the target function values vs. the survey parameters are output in the form of parametric tables (Table 1).

Function: TAKEOFF WEIGHT KG 0
Table dim.: 10 x 10
X-variable: 30/WING LOADING KG/M2 0
Y-variable: 31/WING ASPECT RATIO 2

X / Y	300.	333.	367.	400.	433.	467.	500.	533.	567.	600.
5.00	19322	19071	19066	19249	19606	20121	20819	21748	22933	24423
6.67	19550	19141	19004	19066	19304	19690	20224	20939	21868	23062
8.33	19904	19363	19122	19097	19257	19567	20019	20645	21458	22480
10.00	20322	19663	19333	19238	19336	19590	19992	20558	21306	22257
11.67	20809	20015	19597	19436	19482	19690	20048	20582	21289	22182
13.33	21336	20406	19896	19673	19670	19837	20168	20671	21340	22212
15.00	21914	20832	20227	19941	19892	20025	20328	20798	21453	22320
16.67	22540	21291	20585	20235	20140	20240	20511	20964	21617	22458
18.33	23222	21783	20968	20551	20412	20474	20719	21165	21804	22608
20.00	23962	22310	21376	20889	20702	20724	20951	21388	22002	22775

Table 1. Parametric data table format.

4.1.3. Optimization mode

The application of computer codes for design optimization has become commonplace in many engineering disciplines. The principal advantage, that of reduction in design time, will be illustrated with a simple example[9]. Assume that we have a relatively small analysis program which takes 0.1 CPU sec to evaluate. If we wish to optimize 3 design parameters, and to do so evaluate the analysis program for 10 values for each design parameter, the exercise will take $0.1 \times 10^3 = 100$ CPU seconds. Next, we use a more comprehensive analysis program which takes 10 CPU sec to evaluate. To optimize 10 design parameters in the same way, would require 10^{11} CPU sec $\approx$ 3171 years!

Although the optimizer is generally considered as a "black box" to the engineer, it is useful to have some insight in the mathematical concept of the algorithm to understand its capabilities and limitations. The algorithm included in ADAS is referred to as sequential quadratic programming[10]. The search strategy can best be explained for an optimization problem with two free variables, for which a geometric interpretation can be given in a contour plot, shown in Fig. 10.

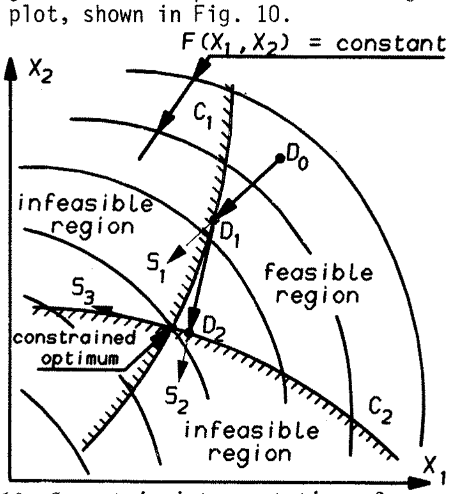


Fig. 10. Geometric interpretation of sequential quadratic programming.

Assume at some iteration step the design is at point D_0, which is feasible but not optimal. The objective function is locally approximated by a multivariable quadratic function and differentiated with respect to the free variables to obtain an estimate for a new, unconstrained optimum. The constraints are evaluated and the design is checked for its feasibility. If feasible, the process is repeated. If infeasible, a line search is performed along S_1 until the design is just feasible. A new quadratic approximation to the objective function and the gradient vector at D_1 are determined. If the gradient vector points toward the feasible region, the constrained condition was due to a prediction error in the quadratic approximation and the process is repeated as described above. If the gradient vector points toward the infeasible region, the design is indeed constrained and the line S_2, tangent to the active constraint C_1, becomes the new, constrained search direction.

After completion of the optimization process, information on the optimum design, including gradients, are output. The user has control over the amount of intermediate information output by the optimizer to monitor the optimization process. Optionally, a convergence history plot can be requested to verify the convergence and to assess the performance of the optimizer (Fig. 11).

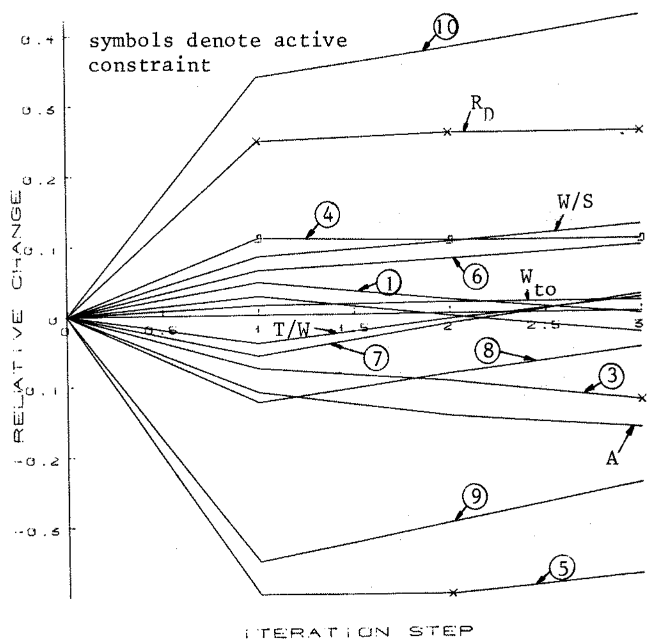


Fig. 11. Optimization convergence history plot.

Multivariate optimization (MVO) studies have been used in aircraft design for more than about 15 years[11,12]. However, applications have shown the results to be strongly dependent on the accuracy and reliability of the predictive model in simulating the underlying physical phenomena[13]: optimization tends to capitalize on deficiencies in the model! Therefore, interpretation and validation of optimization results is still a formidable task and only reserved for experienced designers, assisted by transparent, analytical optimization techniques[14].

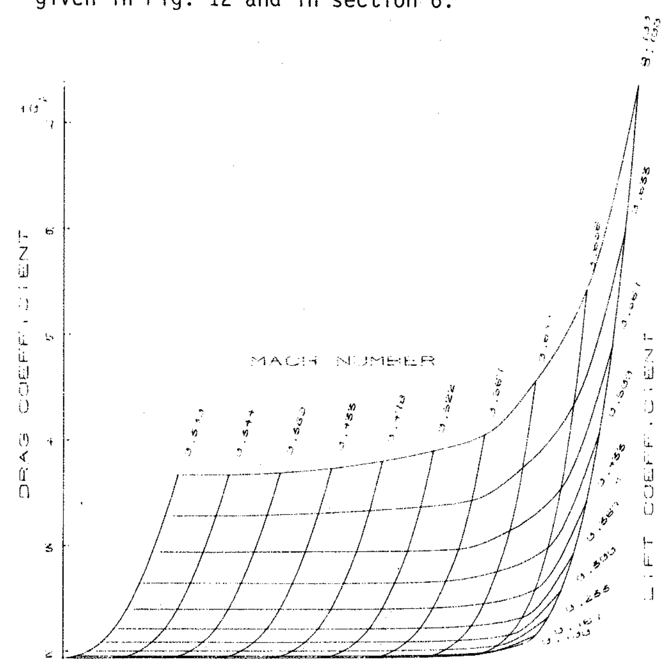
5. DESIGN EVALUATION

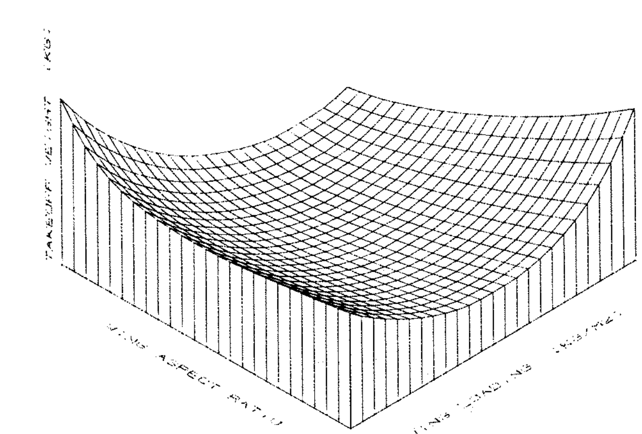
Several options are available under ADAS control to retrieve and display analysis data, output from the ADAP program, for evaluation, i.c.:

- Plots in a plot file can be viewed and, optionally, sent to a remote line-plotter for high quality copies;
- A data file can be browsed through and, optionally, sent to a remote printer for high quality copies;
- Data in a design database can be displayed with simple query commands;
- Data in parametric tables format can be graphically displayed with a data plotting facility;
- The MEDUSA system can be invoked to retrieve and inspect a completed 3-view configuration drawing.

5.1. Data plotting facility

The ADAS graphics facility provides the capability to interactively generate several types of engineering diagrams, conveniently referred to as carpet, surface or contour plots. Examples are given in Fig. 12 and in section 6.





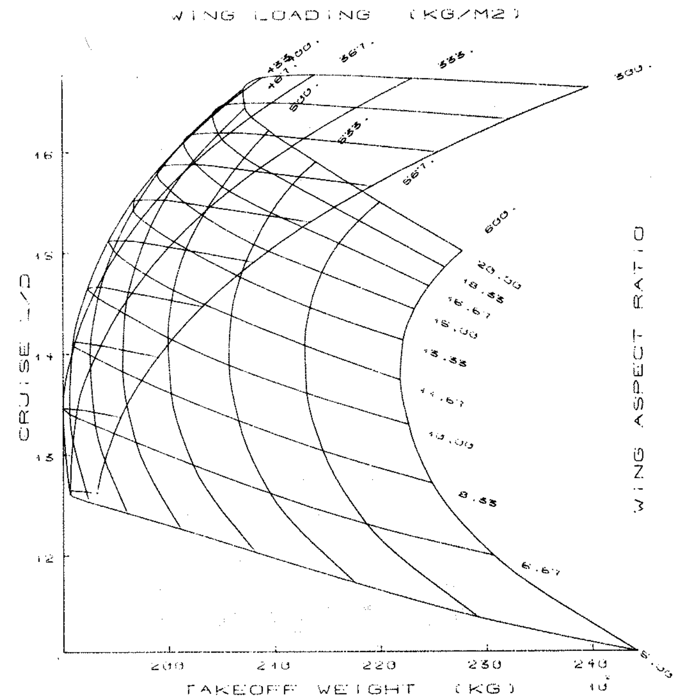


Fig. 12. Parametric data graphical postprocessing.

For each diagram type, specific options are available, e.g. zooming, select curve fitting technique[15], physical plot size, use of cross-hairs to "pick" a design point in a contour plot, etc.

5.2. MEDUSA: geometric modelling and viewing

The ADAS-MEDUSA interface module will automatically modify the MEDUSA drawing, according to geometry changes generated within the ADAP program, e.g. as a result from implicit optimization. In addition, special (link) lines, text and prims will be added to prepare the drawing for input to the solid modeller.

The modelling and viewing operations take place in two separate steps:

- The solid modeller generates a 3-dimensional description, in terms of polygons (tiles), wire lines and faces. The solid modeller provides several model generators, which can be applied for different types of object shapes, e.g. volumes of revolution, ruled surfaces, sweeps and free form modelling. In addition, complicated objects can be modelled with boolean operations, Boolean operations are used to join, subtract or find the complementary volume of, two or more separately modelled, objects;

- The viewer requires a model description and draws a projected solid model image, optionally with hidden lines removed, at a user-specified orientation (Fig. 13).

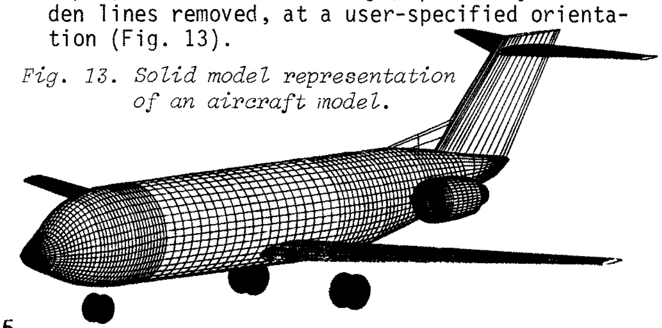


Fig. 13. Solid model representation of an aircraft model.

Alternatively, the shader can be used to generate a shaded image on a raster type graphics terminal (Fig. 14).

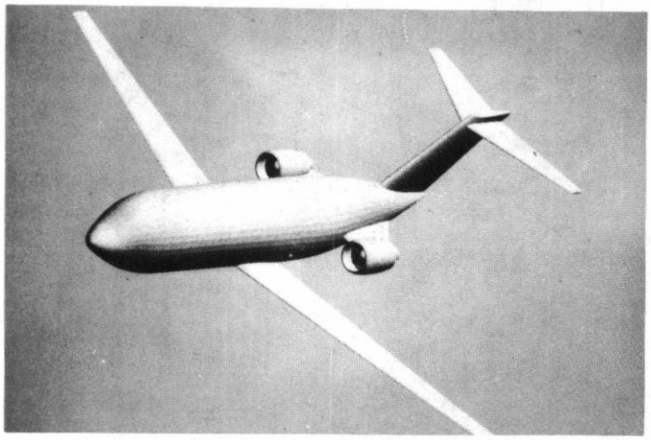


Fig. 14. Shaded image of an aircraft model.

Fig. 15 gives a 3-view configuration drawing representation with some major dimensioning lines added.

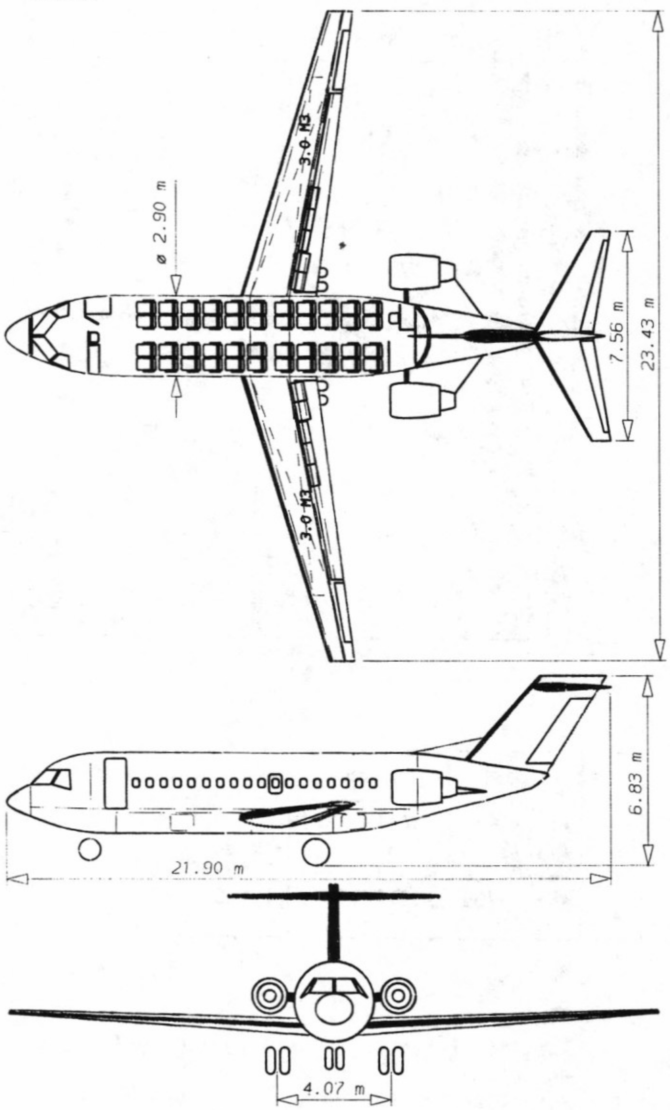


Fig. 15. Final 3-view configuration representation.

This completes one typical design cycle. If the designer is confident that the current design is the "best" possible, it can be submitted to more detailed analysis. For example, the CAPPA program, developed at the DUT and running on an IBM 3083 mainframe, can be used for detailed performance prediction (Fig. 16).

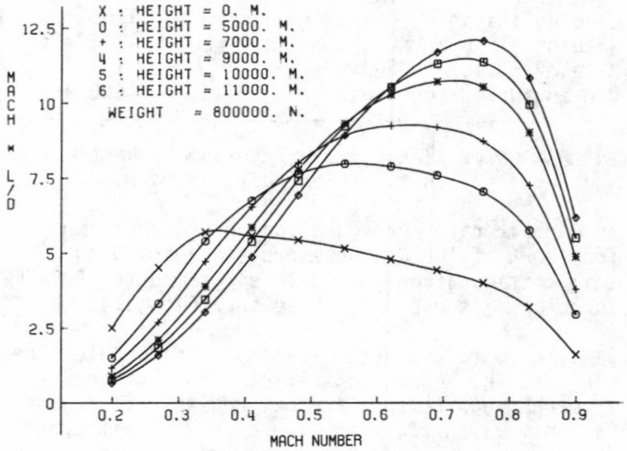


Fig. 16. Detailed performance analysis with CAPPA.

The Department of Aerospace Engineering avails of several computer codes, under custody of different disciplinary teams, specifically for detailed design and analysis[16]. When a suitable DBMS is implemented, attention will be given to the integration of these programs. Here, the MEDUSA system may play an important role as a common tool for manipulation and storage of geometric data. In this respect, the ADAS-MEDUSA interface can prove to be very valuable as a "bridge" to downstream design levels.

6. DESIGN OPTIMIZATION EXAMPLE OF A SMALL PASSENGER TRANSPORT

During their 5th semester, aeronautical students at the DUT can participate in an aircraft design exercise. They can choose from about 25 design specifications in different subsonic aircraft categories, such as general aviation, executive/business, trainers, agricultural aircraft, seaplanes, sailplanes, cargo and passenger transports. Because of limited available time - nominally 200 hrs. - configuration development is necessarily restricted to one design cycle, with little or no optimization aspects involved.

For the purpose of illustration, the ADAS system has been applied to the optimization of a typical aircraft design, based on one such design specification. The major operational and performance requirements are enumerated in table 2. The following mathematical constraints were considered:

1	Cruise engine rating (T/T_{max})	$\leq$	90 %
2	Maximum speed engine rating (T/T_{max})	$\leq$	95 %
3	Maximum fuel equivalent range	$\geq$	3000 km
4	Takeoff field length	$\leq$	1350 m
5	Second segment climb one-engine-out	$\geq$	2.4 %
6	Landing distance	$\leq$	1200 m
7	Rate of climb at sea level	$\geq$	15 m/s
8	Rate of climb at 7500 m	$\geq$	1.5 m/s
9	Rate of climb at 4600 m one-engine-out	$\geq$	0.5 m/s
10	Required/available wing tank volume	$\leq$	100 %

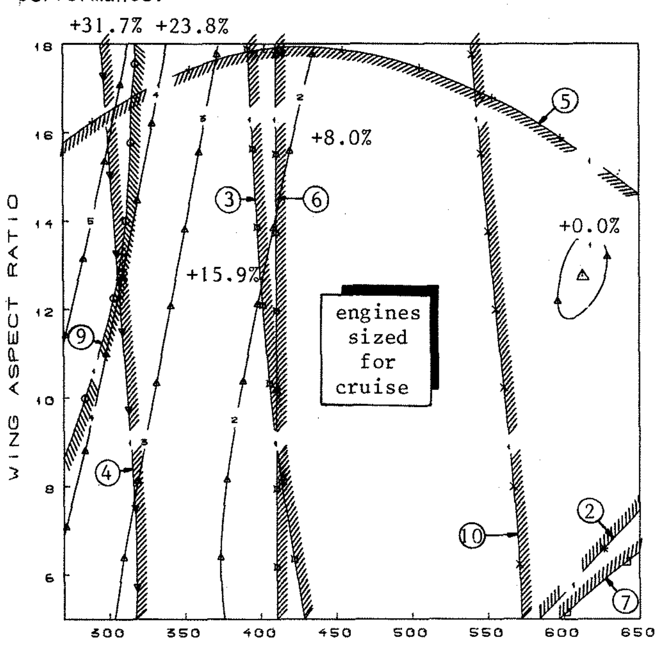
No. of passengers	44
Cruise altitude	7500 m
Cruise Mach number/speed	0.627/700 km/hr
Max. Mach number/speed at 6000 m	0.658/750 km/hr
Max. payload range	1200 km
Max. fuel range	2500 km
Takeoff field length (ISA, SL)	1350 m
Landing distance	1200 m
Rate of climb at SL	15 m/s
Rate of climb at 7500 m	1.5 m/s
One-engine-out ceiling	4600 m

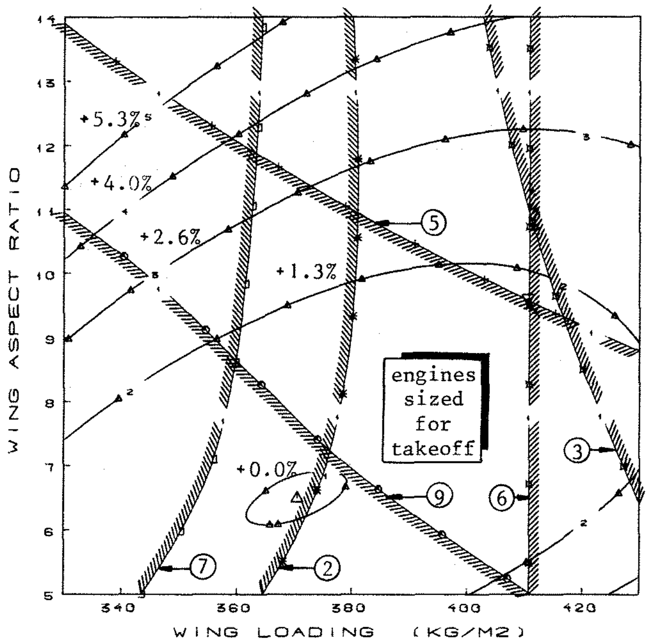
Table 2. Operational and performance requirements for a hypothetical aircraft design.

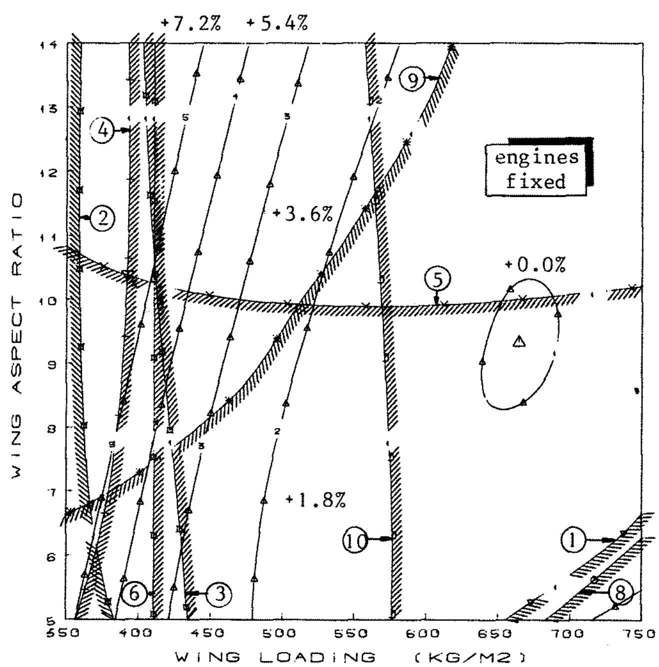
An aircraft configuration developed in the framework of the design exercise, a low wing, T-tail configuration with 2 fuselage-mounted ALF-502 turbofans, is used as a baseline (Fig. 15).

The objective of this example is to determine wing loading and aspect ratio, such that the performance requirements are met and takeoff weight is minimum. In addition, the effect of different engine sizing criteria on the optimum configuration has been investigated. To demonstrate the principles of explicit optimization, 10 x 10 designs were analyzed with different values for wing loading and aspect ratio. The optimum design was selected graphically from a contour plot representation. This procedure was repeated for 3 engine sizing criteria, i.c.: engines sized for cruise, engines sized for takeoff and fixed engines. For each design point, the optimizer was used to match fuel weight to a given maximum payload equivalent range (1600 km) and, where appropriate, vary thrust loading for engine sizing ("rubberizing"). The results from this design optimization study are shown in Fig. 17. Constraints not visible in the indicated design space are not critical.

The case with engines sized for cruise does not result in a feasible design. Due to the relatively low thrust loading, maximum speed and climb requirements cannot be met without derogating field performance.







Case	cruise	takeoff	fixed	∇ Δ con/uncon
W/S	***/ 613	411/ 371	392/ 665	kg/m^2
A	***/12.75	9.57/ 6.54	10.38/ 9.34	
T/W	***/0.236	0.314/0.293	0.293/0.319	
W_{to}	***/16703	19214/19001	19229/17724	kg

Fig. 17. Design characteristics vs. wing loading and aspect ratio.

The time required to analyze 100 designs, including engine sizing and matching fuel load (2 free variables), required about 2.7 hrs CPU and 1.1 min disk access time on the ICI. Turnaround time with an average computer work load, is about

1.5 x CPU time.

In contrast, implicit optimization has been applied, with 4 free variables and 11 constraints, to verify the optimum design, The result was close to the constrained optimum with engines sized for takeoff. This procedure required 11.6 min and 28.7 sec of CPU and disk access time respectively.

In this example, takeoff weight was selected as the function of merit. However, alternative criteria can be considered, e.g. DOC, fuel weight, compound functions, etc., to evaluate a design. Fig. 8 shows the effect of different merit functions on the optimum design.

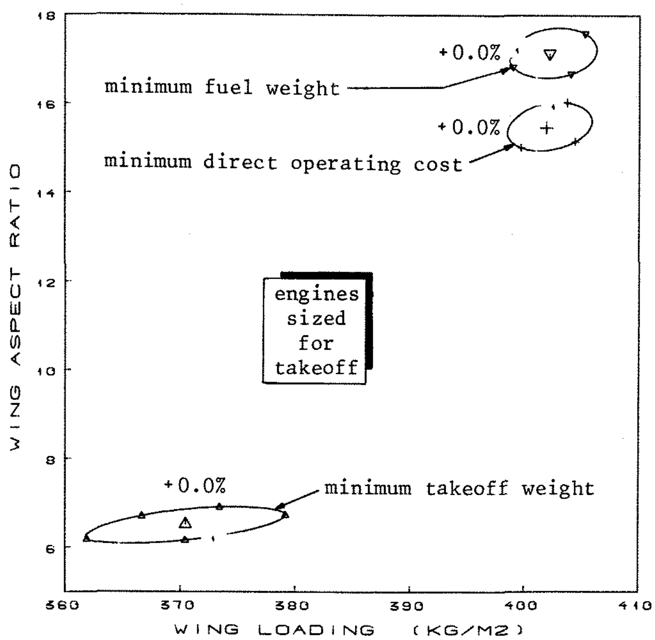


Fig. 18. Effect of different merit functions on the unconstrained, optimum design.

Depending on the merit function, widely different optimum design configurations may result. Hence, the selection of a suitable merit function to assess the design quality, can be critical.

7. CONCLUSIONS AND FUTURE DEVELOPMENTS

A pilot system for computer-aided conceptual aircraft design geared toward a university environment, has been presented. Its principal components and specific features have been highlighted. ADAS has generic properties and capabilities comparable with those of systems in the aerospace industry. It has proven to be flexible and versatile to handle many design problems. The system is experienced as transparent and generally easy to operate.

Enhancement and review of the program library will be a continuous effort. New analysis methods have to be conceived to cover a wider range of design disciplines. These methods have to be sufficiently accurate but with acceptable demand on computer time and memory. Existing modules have to be reviewed and, if necessary, changed or replaced to reflect advances in aircraft design technology or to accommodate unconventional configurations.

Currently, the Department is in the process of introducing CAD/CAM down to the propaedeutical years of the curriculum. Because of the number of students involved, a new computer configuration has been acquired, consisting of 10 SUN-3 single-user workstations, interconnected by a local area network. An additional link will be provided with the ICI. This system will also support the MEDUSA package. In the coming years, ADAS will be converted to UNIX operating system and implemented on SUN. Beyond that point, ADAS will gradually and cautiously be introduced into the preliminary design exercise and offered as a modern tool for aircraft design and optimization.

8. REFERENCES

1. Elias, A.L.: "Computer-aided engineering: the AI connection", Aeronautics & Astronautics, July/August 1983.
2. Stack, S.H.: "Computer-aided design system geared toward conceptual design in a research environment", J. of Aircraft, Vol. 19, No. 2, Feb. 1982.
3. Silver, B.W.: "Optimization studies in aircraft design", SUDAAR NO. 426, Dept. of Aeronautics and Astronautics, Stanford University, June 1971.
4. Torenbeek, E.: "Computer-aided design as a discipline in aeronautical teaching and research", Memorandum M-472, Dept. of Aerospace Engineering, Delft University of Technology, Delft, June 1983.
5. Bil, C.: "ADAS command reference manual", Dept. of Aerospace Engineering, Delft University of Technology, Oct. 1985 (unpublished).
6. Bil, C.: "ADAS database reference manual", Dept. of Aerospace Engineering, Delft University of Technology, Oct. 1985 (unpublished).
7. Wallace, E.: "A computerized system for the preliminary design of commercial airplanes", AIAA paper 72-793, 1972.
8. Torenbeek, E.: "Synthesis of subsonic airplane design", Delft University Press, Delft, 1976.
9. Vanderplaats, G.N.: "Numerical optimization techniques for engineering design", McGraw-Hill, 1984.
10. Gill, P.E., Murray, W., Wright, M.H.: "Practical optimization", Academic Press, 1981, London.
11. Haberland, C., Thorbeck, J., Fenske, W.: "A computer augmented procedure for commercial aircraft preliminary design and optimization", ICAS-84-4.8.1, Sept. 9-14, 1984, Toulouse.
12. Kirkpatrick,D.L.I., Peckham, D.H.: "Multivariate analysis applied to aircraft optimization - some effects of research advances on the design of future subsonic aircraft", RAE TM AERO 1448, Sept. 1972.
13. Vanderplaats, G.N.: "Automated optimization techniques for aircraft synthesis", AIAA paper 76-909, Sept. 1976.
14. Torenbeek, E.: "Fundamentals of conceptual design optimization of subsonic transport aircraft", LR-292, Delft University of Technology, Dept. of Aerospace Engineering, Aug. 1980.
15. Brodlie, K.W.: "Mathematical methods in computer graphics and design", Academic Press, 1980.
16. Torenbeek, E.: "Teaching and learning conceptual design in the CAD era: generating problems and integrating solutions", AIAA paper 85-3081, Oct. 1985.