A98-31582

ICAS-98-4,2,3

AIRFRAME CONFIGURATION DESIGN USING CONSTRAINT PROPAGATION TECHNIQUE CASE WING STRUCTURE

Wahyu Kuntjoro
Department of Aeronautics and Astronautics,
Institut Teknologi Bandung,
Jalan Ganesha 10, Bandung 40132,
Indonesia

Abstract

Design is an iterative process involving synthesis and analysis. The development of airframe design, especially in its early/preliminary stage, demands flexibility in its process. Likewise there are many configuration modifications according to its needs. As a configuration is directly related to structural responses, any changes in configuration need structural analysis to check for new responses. Many design iterations are needed and a lot of time is spent for this. A design tool is needed to support the structural designers to design airframe configuration. The developed tool needs exploiting simple but reasonably accurate analysis methods. In addition, a flexible computer programming technique is needed to handle the required design modifications. Conventional sequential programming is not able to handle innovations effectively, which are needed in a configuration design phase. Symbolic processing techniques such as constraint propagation and object-oriented programming provide the needed flexibility. This paper will describe the development of airframe design methodology, which involves configuration changes. Constraint propagation and object-oriented techniques are utilized in the method development. A computer program has been written which deals with the design of wing structure. Some results of wing design cases will be presented in this paper.

Introduction

Design is an iterative process involving synthesis and analysis. Synthesis is related to creation of structure and includes selection of structural types, configuration, shape and types of component, and the determination of component dimensions. Analysis relates to determination of structural responses such as stresses, strains, and deflections, under external loading. These responses must be within some predetermined allowable ranges.

This paper will describe the development of airframe design methodology, which involves configuration changes. In many of airplane industries, airframe configuration normally is designed through trial and error. Design development, especially in its early stage, demands flexibility in its process. Likewise there are many configuration modifications according to its needs. As a configuration is directly related to structural responses, any changes in configuration need structural analysis to check for new responses. This shows the needs of exploiting simple but reasonably accurate analysis methods. In addition, a flexible computer programming technique is needed to handle the required design modifications.

With respect to design and analysis, the use of computers in aerospace engineering in majority has centered on conventional sequential programs for performing a predetermined series of calculations to analyze a given design. This kind of programming has a fairly limited capability to function only to a design derived from a particular-baseline design. Conceptual design, which includes configuration design, needs a high flexibility. Sequential programming is not able to handle innovations effectively, which are needed in a configuration design phase. Symbolic processing techniques such as constraint propagation and object-oriented programming provides the needed flexibility.

Through object oriented programming, structural tacsonomy can be developed. A wing structure can be regarded as a large entity, which is built from upper-panel, lower-panel, spars, and ribs. This can be viewed as a parent, that is wing, which has several children, those are panels, spars, and ribs. Object oriented programming supports the concept of inheritance, where an object possesses characteristics of its parent object. For example, upper-panel inherits the characteristics of its parent that is wing structure. An objecs also able to interact with other objects.

21st ICAS Congress
13-18 September 1998
Melbourne, Australia

It is known that design of upper panel depends to the rib distance for its buckling behaviour. Design of spars also depends to the rib distance as well. On the other hand, rib distance directly influences the loading of ribs, hence the design of ribs. From the example above, it is clear that there are common parameters, which influence the outcome of the design of more than one component. This kind of interaction can be simulated well by object-oriented programming and constraint propagation technique.

The development of airframe design methodology, which involves changes in configuration, is described in this paper. Constraint propagation and object-oriented techniques are utilized. A computer program has been written which deals with the design of wing structure. Some results of wing design cases will be presented in this paper.

Constraint Propagation Technique

Conventional programming consists of sequences of explicit instructions for performing calculations. Constraint propagation(1), on the other hand is a technique, which provides mechanisms for transforming declarative statements of mathematical relationships into their underlying imperative forms. For example, in the subject of structural design a statement

$$S=F/A \qquad (1)$$

computes the stress of a rod, S, as the division of the force, F, over cross sectional area of the rod, A, figure-1.

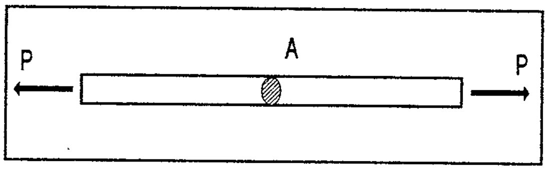

Figure-1 A rod of cross-sectional area A, loaded by an axial force P.

Parameter S can also be regarded as material strength. However, the above statement is useless if we wish to find the required area, A, based on the material strength S. It is necessary to modify the above statement into,

$$A=F/S \qquad (2)$$

If a computer program for stress calculation has been written, this change of intention in analysis implies that the program must be modified and rewritten.

In airframe preliminary design phase, many design modifications, including changes of configuration, are performed and a lot of calculation must be done. Sequential programming is not able to handle design modification effectively, which are needed in a configuration design phase. Symbolic processing techniques such as constraint propagation and object-oriented programming provide the needed flexibility to deal with requirements in early design phase.

Constraint propagation technique is based on a concept that for a parameter, in addition to a value, a set of value supplier is provided:

No-Value: The parameter currently has no value.
User: The current value of the parameter is supplied by the user.
Computed: The current value of the parameter is computed by a constraint.

The value supplier is either set to 0 (FALSE) or 1 (TRUE). If any of value suppliers is set to TRUE, then the other two value suppliers are set to FALSE.

With respect to our problem stated in equation (1), the design statement "S=F/A" is treated as an imperative statement which relates three parameters, those are stress (or strength) S, external force F, and cross sectional area A. To each parameter, a set of value supplier is attached. The design statement itself is considered as a constraint. At any stage of design, a parameter has only one of its value supplier set to TRUE, and the other two value suppliers set to FALSE.

The "S=F/A" constraint will be executed if two of its parameters have No-value=0 (which means they have value). For example, if a material has been selected and its strength is known, then a certain stress level is specified, User=1. If the working force is known, either from other calculation (Computed=1) or supplied by the user (User=1), then the constraint will be executed as "A=F/S". As has been mentioned, the force parameter F can be the result of other calculation performed in other constraint. This constraint for example, can be the constraint which relates the couple forces with its moment and the moment arm.

In a design problem which is handled by constraint propagation approach, whenever there is a change of a parameter's value supplier, it notifies this change to all of its constraints. In this way, constraint propagation is triggered recursively. Hence a modification to a certain part of structure automatically resize all other structural components.

Object-Oriented Programming

Object-oriented programming is a computer program written in the form of data abstraction⁽²⁾. Each module in an object-oriented architecture is developed under the principles of data abstraction. These principles are used for the development of problem solution without resorting to complex detail interactions among various components in the program.

Modules in an object-oriented system are called *classes*. Object is a class representation or class instance. A class structure includes data characteristics and the behaviour of objects it defines. In other words, a class is the template of object(s). For example, executive airplane is a class (of airplane), while Gulfstream IV and Cessna Citation are examples of objects which represent that class.

An object has unique characteristic, which makes it different to other objects. An object also possesses a capability to operate to data/characters it possesses. If needed, an object could interact with other objects.

An object-oriented system also supports concept of inheritance. Inheritance concept allows objects to be set up in tacsonomy where a special object possesses characters and operations inherited from a more general object. However, objects can redefine inherited characters and operational capabilities. For example, fighters can be regarded as a general class while ground-strike-planes and air-superiority-planes are more specialized classes. On the other hand, fighter can be regarded as a more specialized class of military-plane class.

For our wing design problem, wing can be regarded as a general class. This class has several child-classes, those are, upper-panel, lower-panel, front-spar, rear-spar, and rib(3,4). This is shown in figure-2.

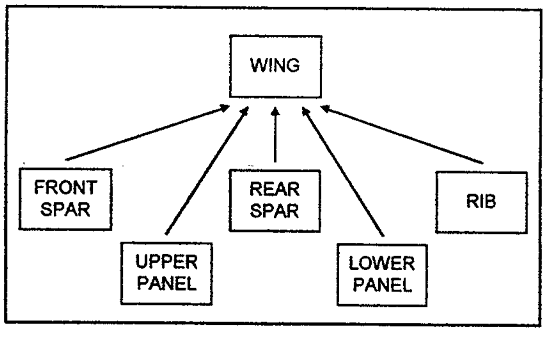

Figure-2 Tacsonomy of wing structure.

Each class defines an object. Each object possesses design parameters and constraints unique to its class. Every parameter is a given a set of value supplier. A change of a parameter's value supplier is notified to all of its constraints. The constraint then propagates and design modification is performed.

Design Requirement

Currently, the design requirement is based on static analysis. The wing to be designed must able to carry the static load without failure. The wing is based on the integrally machined wing. The analysis is based on the wing-box and needs overall wing-box geometry and the external load in the form of Shear Force, Bending Moment, and Torsion distributions. The analysis is performed to each rib-bay, hence the geometry of each rib-bay is required.

The design criterion for the wing-box in general is its stiffness distribution. This covers both the bending stiffness and torsional stiffness. The stiffness requirement will limit the lateral deflection and rotational twist. The stiffness distribution will also affect the aeroelastic behaviour of the wing.

The upper panel is designed not to buckle under compression load. The panel should also carry the load without material failure. The user can determine the geometry of the upper panel. The stiffener pitch is set to be constant.

The lower panel is designed not to fail under tension load. In addition to that, the panel must able to carry 1-g stress without fatigue damage. However, the value of maximum fatigue stress level for 1-g load is not specified in the program. Similar to the upper panel, the stiffener pitch of the lower panel is set to be constant.

Both front spar and rear spar are designed not to buckle under shear load. In this way, the spar concept is based on shear resistant web. In addition to that, the shear stress must be lower than the shear strength of the material used. The user can specify number of spar's stiffeners (or uprights) between ribs at every rib-bay.

Similar to spar, the rib is designed not to buckle under shear load. The shear stress must be lower than the shear strength of material used. For each rib, the user also able to specify the number of uprights between front spar and rear spar.

For every wing component, that is upper panel, lower panel, front spar, rear spar, and ribs, the user can specify the material used. However the prototype developed works only with isotropic material.

Object Implementation

Based on the classification as shown in figure-2, object classes are developed. These classes are WING, UPANEL (upper-panel), LOPANEL (lower panel), FSPAR (front spar), RSPAR (rear spar), and RIB (rib). Each object contains design parameters and constraints related to the associated object. Constraints refer to the analytical equations, which are function of design parameters. These equations are either related to geometry operations or design and analysis of the associated structural object. The structural analysis for wing structure is taken from several literature(5,6,7).

Figure-3 shows geometry of the wing-box(8). The wing box is divided into several rib-bays, which can be specified by the user. The wing will have the tip-rib and root-rib. The rib distance needs not to be uniform. The user can specify the box dimensions at each rib-bay. At present, the loading is given as distribution of shear force, bending moment, and torsion at each rib-bay. During the program run, the user can change the loading, number of rib-bay, and box dimension, and look at the response of the modified structure.

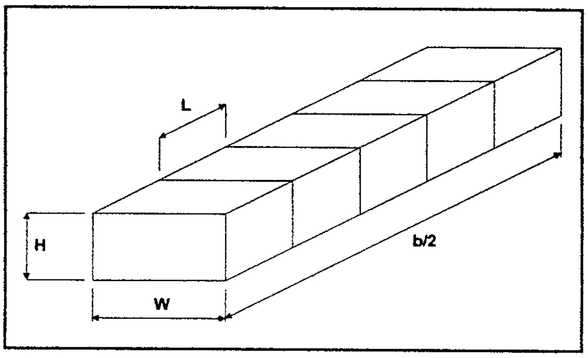

Figure-3 Geometry of the wing-box.

The program structure of WING object-class is shown in figure-4. In accordance to the tacsonomy shown in figure-2, the object-class WING is a top-class (or parent class). As the parameters are set to be **general** parameters, all parameters are inherited by all of its **child**-classes. The listing of parameters and constraints by no means is complete. However it will give the idea of how an object is programmed. Every parameter is provided with a set of value supplier. During run-time, an object is defined of this class, which is named as *wing*.

Class wing:
parameters:
b (wing span), *w[i]* (wing box width), *H[i]* (box height), *L[i]* (rib distance), *n* (number of ribs), *F[i]* (axial force at a section), *V[i]* (shear force at a section), *M[i]* (bending moment at a section), *T[i]* (torsion at a section), ...
constraints:
axial_force() $F[i]=M[i]/H$
controller()
....

Figure-4 Program structure of object class WING.

Geometry of the upper-panel is shown in figure-5. At present, the panel is assumed to be an integral panel with blade stiffener. As shown in figure-5, the stiffener pitch is constant. For analysis, the spar caps are assumed not contributing to the panel capability in transferring the load. Figure-6 shows the program structure of UPANEL object-class, which is the representation of upper panel. This class defines an object called *upanel*.

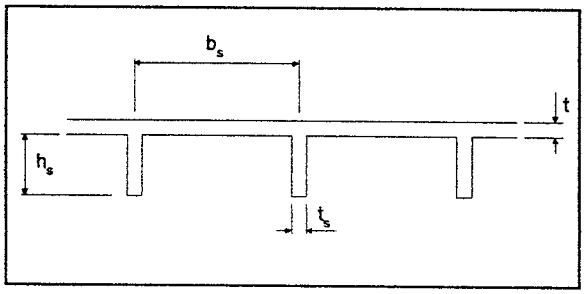

Figure-5 Geometry of the upper panel.

Class UPANEL:
parameters:
t[i] (panel thickness), $h_s[i]$ (stiffener height), $t_s[i]$ (stiffener thickness), $b_s[i]$ (stiffener distance), *t_eff[i]* (effective thickness), *A[i]* (panel area), *sig[i]* (stress),
.....
constraints:
eff_thickness() $t\_eff=t+(h_s*t_s)/b_s$
panel_area() $A=(w*t\_eff)$
panel_stress() $S=F[i]/A$
....

Figure-6 Program structure of object class UPANEL.

The object as shown in figure-6 also contains margin of safety calculation with respect to the material strength and buckling.

Geometry of the lower panel is defined similarly to the upper panel. The same also applies to the stress response. However, for the strength criteria, the buckling of panel is omitted. This is in an assumption that lower panel is not designed for compression. Lower panel is also required to transfer the 1-g load for a certain stress level, which is based on fatigue consideration.

Geometry of the spar (front spar as well as rear spar) is shown in figure-7. The spar geometry is defined based on its height, web thickness, and number of uprights at every rib-bay. The upright pitch is defined as constant at every bay. The spar is designed as non-shear-buckling beam. The program structure of object-class FSPAR (front spar) and object-class RSPAR (rear spar) are similar. FSPAR and RSPAR define objects called *fspar* and *rspar* respectively. Class-object RSPAR is shown in figure-8.

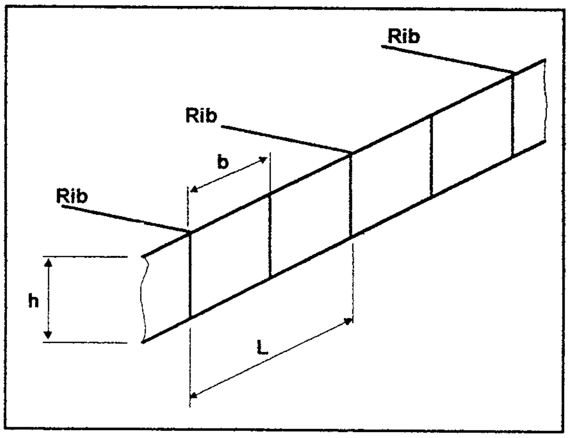

Figure-7 Geometry of Rear Spar.

Class RSPAR:
parameters:
ns[i] (number of upright), *t[i]* (web thickness), *h[i]* (spar-beam height), b*[i]* (upright distance), *tau[i]* (shear-stress), *tauloc[i]* (local buckling), ...
constraints:
upright_pitch() $b_s=L[i]/(ns[i]+1)$
shear_buckling() $tauloc[i]=K[i]*E[i]*(t[i]/d[i])^2$
MS_buckl() $MS\_loc[i]=tauloc[i]/tau[i]-1$
MS_mat() $MS\_y[i]=ssy/tau[i]-1$
....

Figure-8 Program structure of object class RSPAR.

Figure-9 shows geometry of the rib. The geometry is defined based on its height, web thickness, and number of uprights at every rib. The upright pitch is defined as constant. Similar to spar, rib is designed as non-shear-buckling beam. The program structure of class-object RIB is shown in figure-10. During program-run, an object called *rib* is defined of this class.

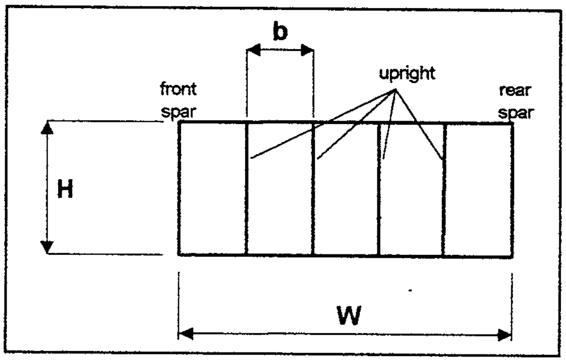

Figure-9 Geometry of rib.

Class RIB:
parameters:
ns[i] (number of upright), *t[i]* (web thickness), *h[i]* (rib height), b_s*[i]* (upright distance), *Fr[i]* (rib load), *tau[i]* (shear-stress), *tauloc[i]* (local buckling), ...
constraints:
rib_load() $Fr[i]=(V[i]-V[i-1])/w[i]$
shear_buckling() $tauloc[i]=K[i]*E[i]*(t[i]/d[i])^2$
MS_buckl() $MS\_loc[i]=tauloc[i]/tau[i]-1$
MS_mat() $MS\_y[i]=ssy/tau[i]-1$
....

Figure-10 Program structure of object class RIB.

Case and Result

The computer program is written in a Pentium Personal Computer using C++ language. Trials have been performed to the prototype for wing design cases. Two design-scenarios will be presented in this paper. Both involve the same hypothetical wing structure.

Case-1

A (half) wing-box has semi span of 5000 mm and 10 rib bays each of 500 mm long. The wing is of constant chord type and the width of wing box is 600 mm. The (average) height of wing box is 200 mm. Each panel (upper and lower) has 5 stiffeners, excluding spar flanges, having constant pitch. The spar web is stiffened by one upright between ribs.

Each rib is stiffened by 3 uprights of uniform pitch. With exception of ribs, the thickness of all components (skin/web, stiffener/upright) is 1.6 mm from rib-bay 1 to rib-bay 5 and 2 mm from rib-bay 6 to rib-bay 10. The height of stiffeners/uprights in panels, spars, and ribs are 30 mm. The thickness of web and upright of ribs are 1.2 mm. The material for upper panel is Al-7075 T6. All other components are made of Al-2024 T3 material.

The wing box is required to carry the load (limit) distribution given in Table-1. It is assumed there is no torsion.

Table-1 Loading

Rib bay	Shear Force V (N)	Bending Moment M (N.mm)
1	1250	312500
2	2500	1250000
3	3750	2812500
4	5000	5000000
5	6250	7812500
6	7500	11250000
7	8750	15312500
8	10000	20000000
9	11250	25312500
10	12500	31250000

Margin of safety (MS) of various wing components are shown in Table-2. The MS shown are the most critical MS condition for the associated components. For upper-panel, rib-bay next to the wing-root is critical due to general buckling at ultimate load. From the MS values it can be concluded that the wing is safe from static point of view.

Table-2 Margin of Safety (MS) of Case-1

Rib-bay	Upper panel	Lower panel	Front spar	Rear spar	Rib
0					25.9
1	52.3	238.6	17.7	17.7	7.97
2	12.3	58.9	8.33	8.33	4.38
3	4.92	25.6	5.22	5.22	2.84
4	2.33	14.0	3.67	3.67	1.99
5	1.13	8.58	2.73	2.73	1.44
6	1.75	7.32	5.07	5.07	1.07
7	1.02	5.11	4.21	4.21	0.79
8	0.55	3.68	3.56	3.56	0.58
9	0.22	2.70	3.05	3.05	0.42
10	0.0	2.00	2.64	2.64	0.34

There are 11 ribs including tip-rib and root-rib. However the rib calculation has not included the crushing effect due to bending, and loading redistribution at root.

Case-2

In the second case, it is required to reduce the wing thickness into 180 mm. To counter the resulting reduction of box-volume, the box width is increased to 640 mm. The other configuration and dimensions remain the same. The MS of the new configuration is shown in Table-3.

Table-3 MS of Case-2

Rib-bay	Upper panel	Lower panel	Front spar	Rear spar	Rib
0					22.8
1	43.3	225.7	18.5	18.5	6.94
2	10.1	55.7	8.73	8.73	3.76
3	3.92	24.2	5.49	5.49	2.40
4	1.77	13.2	3.87	3.87	1.65
5	0.77	8.07	2.89	2.89	1.17
6	1.40	6.87	5.33	5.33	0.83
7	0.77	4.78	4.43	4.43	0.83
8	0.35	3.43	3.75	3.75	0.40
9	0.07	2.50	3.22	3.22	0.25
10	-0.13	1.83	2.81	2.81	0.19

As shown in Table-3, near to the wing-root, the MS of upper-panel is negative. In fact the negative margin is due to the local buckling. Hence the effect of changing the wing overall geometry can be found using this program.

Conclusion

A design tool, which is taking into account the change of configuration, has been developed. The developed tool exploits simple but reasonably accurate analysis methods. In addition, a flexible computer programming technique is needed to handle the required design modifications. Conventional sequential programming is not able to handle innovations effectively, which are needed in a configuration design phase. Symbolic processing techniques such as constraint propagation and object-oriented programming provide the needed flexibility. Hence, the system is developed based on the concept of constraint propagation and object-oriented methodology.

The constraint propagation provides a set of value supplier to parameters, which are involved in the design process. Design constraints are set-up as functions of these design parameters. By providing values to these sets of parameter's value supplier, constraints are triggered recursively. A computer program prototype has been written in C++ language on a Pentium PC. The program has been tested for various cases of wing structural design.

From the trials conducted, the developed tool proves to be flexible and capable to handle the innovations required during the configuration design phase. Hence the use of constraint propagation and object-oriented methodology proves capable to deal with this type of design problem.

The analysis performed in this prototype is simple. The wing-box is assumed to be of rectangle shape omitting the airfoil shape. However, this is regarded as reasonable in the early phase of design. Currently, the rib design has not taken into account the crushing load. Local effect such as load redistribution at wing-root and local load introduction have not been implemented. More detailed analysis will be implemented in the system. Following this, the system will be extended to cover the conventional riveted panel as well.

It needs mentioning that the system developed is capable to produce a feasible configuration design. The component sizes of this feasible configuration can be optimized using an optimization program(8).

References

1. Kolb, MA.
Constraint-Based Component-Modeling for Knowledge Based Design. AIAA 92-1192

2. Booch, G.
Object Oriented Design with Applications. The Benjamin/Cummings Publishing Company, Cal., USA, 1991.

3. Kuntjoro, W.
Constraint Based Approach for Design of Wing Structure. IASE-7 Conference Proceeding, Manchester - UK, 1995.

4. Kuntjoro, W.
Object Oriented Knowledge-Base for Airframe Design. Artificial Intelligence'95 Proceeding, PIKSI-ITB, Indonesia, 1995.

5. Bruhn, E F.
Analysis and Design of Flight Vehicle Structures. Tri-State Offset Co., USA. 1965

6. Niu, M C Y.
Airframe Structural Design. Conmilit Press Ltd., Hong Kong, 1988.

7. Kuntjoro, W.
Comparison of Classical Analysis to Structural Test Result for Airframe Design Use. SPP-DPP Report No. 17523196, ITB, Indonesia, 1998.

8. Kuntjoro, W
Airframe Configuration Design using Constraint Propagation Technique and Optimization Method. RUT Annual Report 1998, National Research Council, Indonesia, 1998.