COMPUTER-AIDED DERIVATION OF EQUATIONS OF MOTION FOR ROTARY-WING AEROELASTIC PROBLEMS

F. Kießling
DFVLR-AVA Institute of Aeroelasticity
Göttingen, F. R. G.

ABSTRACT

A general computer-algebra system has been applied to derive literal equations of motion for the aeroelastic behaviour of rotary-wings. Inertia, elastic, structural damping, aerodynamic, and gravitational contributions are considered. Modal degrees of freedom are provided to represent elastic rotor blades. The program input comprises mainly a kinematic description of the system. A weighting scheme is used to obtain the most important terms in a consistent manner. Multiblade coordinate transformation is applied to reduce or to eliminate periodic coefficients. As output, matrices are written in FORTRAN code, which reflect the mathematical model and can be used for further numerical calculations. As an example, the suggested procedure is applied to a model of a two-bladed wind turbine mounted on an elastic tower.

1. INTRODUCTION

The theoretical investigation of aeroelastic stability and dynamic response of rotary-wing aircraft requires considerable effort in mathematical modeling. Compared to the fixed-wing case, the inertia forces must now include centrifugal and coriolis terms. The aerodynamic situation of the problem is even more difficult due to the complex unsteady motion of the rotating lifting surfaces. In some instances, gravity plays a role in the aeroelastic behaviour of the system. And last but not least, nonlinear effects must be considered for a number of aeroelastic problems. Structural nonlinearities are important for the modeling of modern hingeless or bearingless rotor designs. Nonlinear aerodynamic effects occur in the separated or transonic flow regimes entered by the rotor blades. It is not intented here to deal with the state-of-the-art aeroelastic modeling of all these physical phenomena, which has been addressed recently in /1/. Only the general problem of deriving equations of motion from a basic kinematic description will be treated.

The amount of labour spent for algebraic manipulation increases considerably as more detailed rotor blade and hub descriptions or rotor/body-coupling are taken into account. The manual derivation process becomes increasingly tedious and error-prone, although the procedure is straightforward. So, only relatively simple models can be handled in this way. For more sophisticated models, it appears necessary to start with their numerical treatment very early. For this reason, two different categories of mathematical models for rotary-wing aeroelastic problems can be identified in the literature:

1. Simplified models
This type, characterized by a small number of degrees of freedom and a large number of simplifying assumptions, is necessary to gain insight into specific problems, which perhaps have been already recognized. The amount of manual derivation is kept within acceptable limits. A typical example of this category is the rotor whirl-flutter model described in /2/. It has been observed that numerous algebraic simplifications can be carried out, especially with the introduction of multiblade coordinates. This results in exact and efficient analytical formulations for subsequent numerical analysis, especially for parameter studies. An additional advantage of simplified models lies in the fact that the structure of the equations becomes visible by their analytical rather than numerical representation and the effect of parameters entering the problem can be estimated by their appearance in literal form. Of course, the scope of such models is limited and a complete re-derivation is usually necessary for incorporating additional features into the model.

2. Comprehensive models
Due to the large number of design parameters of a rotary-wing aircraft, complex general-purpose models are very important for qualification of the system as a whole. The state-of-the-art of this category was recently documented in /3/. This model type is characterized by a large number of degrees of freedom and sophisticated aeroelastic modeling capabilities. As already mentioned, the equations of motion of such models cannot be derived manually. They are put into numerical form in an early stage and the bulk of work is left to the number-crunching capabilities of a fast computer with large capacity. Usually, such models are incorporated in the form of a time-domain simulation of a nonlinear system of differential equations. For stability information, the relevant matrices of a linearized model can be obtained only in numerical form. This involves numerical differentiation for calculation of the Jacobian matrices, a process which can cause problems. Due to their general-purpose character and size, comprehensive models tend to be less efficient for parameter studies. Numerous expensive computer runs will be necessary to gain experience and confidence in handling of such global models.

The question arises as to whether it is possible to attack the problem of derivation of literal equations of motion for aeroelastic problems of rotary-wings with the aid of symbolic manipulation capabilities of computers. By this way, some of the advantages of manually derived models are retained in more complex cases and the numerical process with its associated rounding errors is deferred. The principal differences between the discussed approaches are shown in figure 1.

2. COMPUTER ALGEBRA

Symbolic mathematical computation, or computer algebra for short, is the use of computers for performing symbolic mathematics such as algebra and calculus. Up to now, its application in aeronautical science and engineering is not wide-spread. Instead, computers are mainly used to perform numerical work. In celestial mechanics, general relativity, and high-energy physics, for example, algebraic manipulation on computers has been applied successfully to solve complex problems with too lengthy manual derivation.

A number of general-purpose symbolic program systems has been developed during the last two decades /4/. Most of them are based on special programming languages for list and string processing like LISP, which are unknown to many users of digital computers and require a large portion of storage according to their various algebraic capabilities. To facilitate interactive operation, programs are translated often by an interpreter rather than a compiler, but this is considerably slower. Execution times of symbolic programs written by an inexperienced user may sometimes appear discouraging. For these reasons, it seems attractive to develop special-purpose algebraic processors written in a more popular language like FORTRAN. Symbolic Newton/Euler equations of motion were derived for a system of connected rigid bodies on a FORTRAN basis using index coding /5/. For generation of helicopter equations of motion a symbolic processor has been written in FORTRAN and applied to a rotor/body model with more than two centrally hinged rigid blades, dynamic inflow, and in forward flight conditions, /6/.

Up to now, the better choice between general-purpose systems and special-purpose packages is not evident. Some program features, which may seem unnecessary at first view, will become welcome tools after some time spent with actual program development. Thus, rather than to write a new FORTRAN program with algebraic manipulation capabilities, which must be checked itself, it was decided to use an already available general-purpose language. Its features should be exploited as far as can be expected from an "average" user and not from a computer-algebra specialist with thorough knowledge of the underlying system.

Initial experience with computer-algebra applications in the field of rotary-wing dynamics has been gathered with the derivation of finite element matrices for a rotor blade element /7/. This was done first with the PL/1-FORMAC language /8/. Later on, it was found that another available computer-algebra system, REDUCE 2, was more suited to some desirable operations, like matrix algebra, see /9/. Therefore the following work was continued on this LISP-based system. REDUCE syntax is like ALGOL, so it is not very difficult to become acquainted with. At DFVLR/AVA Göttingen the REDUCE version of August 15, 1979 is implemented on a SIEMENS 7.865 computer and requires a minimum of about 300,000 Bytes storage. It can be used in batch and interactively under TSO. The latter mode is a very powerful tool for testing purposes. REDUCE provides capabilities for

1. simplification, expansion, substitution, and analysis of algebraic expressions,
2. indefinite-precision integer, rational, and floating-point numbers,
3. different output style including FORTRAN-compatible format,
4. differentation, integration, and complex arithmetic,
5. matrix algebra,
6. recursive procedures,
7. check/restart

among others. REDUCE is written in a special mode of its own language bearing more resemblance to LISP. For illustrating the REDUCE language and some of its features, a small self-contained program is shown in figure 2, where the derivation of the equation of motion of a simple plane pendulum is performed.

3. EQUATIONS OF MOTION

Several methods can be applied to derive equations of motion and will give the same answer if the same assumptions are introduced. Well-known examples are the equilibrium method, Lagrange's form of D'Alembert's principle, and Lagrange's equations. Supposed differences in the results must be attributed to different choices of unknowns. In Lagrange's equations only true generalized coordinates are permitted, but it is also possible to use more favorably some other primary unknowns such as so-called quasi-coordinates in alternate methods /10/. For manual derivation of equations of motion of a specific system it is often advantageous to separate subsystems judiciously, to introduce constraint forces, and to apply equilibrium equations including inertia forces. But then the constraint forces must be eliminated to obtain a set of equations of minimum order. To generate equations of motion by computer algebra, it is desirable to choose an "automatic" method requiring a minimum amount of user input and interaction. The procedure should be straightforward for a relatively general class of models. Therefore in the present case, generalized coordinates and their time derivatives are considered to be of primary interest and Lagrange's equations together with the principle of virtual work has been chosen:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \underline{\dot{q}}}\right) - \frac{\partial T}{\partial \underline{q}} + \frac{\partial U}{\partial \underline{q}} + \frac{\partial D}{\partial \underline{\dot{q}}} = \underline{Q} \qquad (1)$$

where $\underline{q}$ contain the generalized coordinates, T is the kinetic energy, U is the elastic potential energy, D is the dissipation function, and $\underline{Q}$ contain the generalized forces not covered by the preceding contributions and which are derived from virtual work. An additional advantage of (1) compared with the equilibrium method is the ease with which modal degrees of freedom can be incorporated in an early stage of development. Modal degrees of freedom, especially corresponding to eigenmodes of various subsystems, enjoy a very simple formulation of the linear elastic contributions in the form of generalized stiffnesses in the equations of motion. So, if eigenfrequencies, eigenmodes, and generalized masses of a substructure are known from prior calculations and/or modal survey tests and modal degrees of freedom are chosen, nothing more must be known about the details of the elastic behaviour.

The direct symbolic computation of (1) with REDUCE was tried first, but succeeded only for small systems like a simple model for helicopter ground resonance instability. With increasing model complexity an error message well known to users of computer-algebra systems indicated that storage capacity was exhausted due to "expression swell." Counteraction of providing larger regions of (virtual) storage was not very successful, because the processing of large expressions like the Lagrangian function was prohibitively slowed down due to enormous paging activities of the virtual storage system. Also the expected expressions are very lengthy, because they contain a whole equation or at least a large part of it. Obviously it would be better to compute only symbolic expressions for elements of matrices, which occur in a general structure of the equations of motion. Thereby, the handling of large expressions is avoided and only a relatively small amount of computer storage is necessary. For this aim, the original equations (1) must be processed further to obtain a matrix structure. The general form of the matrix elements is then programmed to evaluate symbolic expressions.

In general, rotary-wing aeroelastic models must account for nonlinear behaviour. To investigate stability it is normal practice to linearize the nonlinear equations about some reference state and to study the linear perturbation equations. Thus the task involves two steps, first, the determination of the reference state and, second, the solution of the perturbation equations which may be a function of this reference state. In the following it is assumed that the reference state is given. If the reference state value does not enter the perturbation equations, the corresponding generalized coordinate is called linear. For applications up to now, it was assumed that the systems can be treated for the most part with linear generalized coordinates $\underline{q}_1$, but also some generalized coordinates $\underline{q}_n$ exist, which cannot be linearized anyway (the rotor azimuth angle, for example). Accordingly, two sets of equations must be generated, which are coupled with each other. In the following a condensed description is given to obtain the matrix structure of (1). For brevity the derivation of equations for determining $\underline{q}_n$ will be omitted, but in fact this part has been programmed with computer-algebra also.

3.1 PRELIMINARIES

Inertial coordinates of a generic point of a rotor blade are represented as

$$r_i = r_i(\underline{q}_1, \underline{q}_n) \quad . \tag{2}$$

In the following, elements of $\underline{q}_1$ are subscripted with italic letters. Column matrices and square matrices are denoted by a single and double underscore, respectively. i,j,k,l are used for subscripts denoting Cartesian coordinate directions. The summation rule for corresponding subscripts is applied.

3.2 INERTIA FORCES

Inertia forces are derived from the kinetic energy contributions in (1). The coordinates of a generic mass element on a rotor blade in an inertial system is given by

$$r_i = r_{i0} + J^r_{in}\, q_n + \frac{1}{2} H^r_{ikn}\, q_k\, q_n + \cdots \tag{3}$$

with i = 1, 2, 3. Expansion into a Taylor series has been applied with respect to $\underline{q}_1$. To obtain consistent linear equations it is necessary to retain all terms in $\underline{q}_1$ up to the second order.

$$r_{i0} = (r_i)_0 \tag{4}$$

$$J_{in} = \left(\frac{\partial r_i}{\partial q_n}\right)_0 \tag{5}$$

and

$$H_{ikn} = \left(\frac{\partial^2 r_i}{\partial q_k\, \partial q_n}\right)_0 \quad , \quad H_{ikn} = H_{ink} \tag{6}$$

are the inertial coordinates and elements of the Jacobian and (symmetric) Hessian matrix, respectively, evaluated at the reference state of $\underline{q}_1$. Taking the time derivative of (1) and squaring the result yields the kinetic energy contribution of the mass element per unit density. This expression must be integrated over the blade volume. Putting the derived kinetic energy into (1) results in the following matrix form

$$M^K_{mn}\, \ddot{q}_n + \left(D^K_{mn} + G^K_{mn}\right) \dot{q}_n + \left(K^K_{mn} + N^K_{mn}\right) q_n = Q^K_m \quad , \tag{7}$$

for the inertia contributions (superscript K), where

$$M^K_{mn} = \int J^r_{im}\, J^r_{in}\, dm = M^K_{nm} \quad , \tag{8}$$

$$D^K_{mn} = \dot{M}^K_{mn} = D^K_{nm} \quad , \tag{9}$$

$$G^K_{mn} = \int \left(J^r_{im}\, \dot{J}^r_{in} - \dot{J}^r_{im}\, J^r_{in}\right) dm = - G^K_{nm} \quad , \tag{10}$$

$$K^K_{mn} = \int \left(\ddot{r}_{i0} H^r_{imn} - \dot{J}^r_{im} \dot{J}^r_{in} \right) dm + \frac{1}{2} \dot{D}^K_{mn} =$$

$$= K^K_{nm} , \qquad (11)$$

$$N^K_{mn} = - \frac{1}{2} \dot{G}^K_{mn} = - N^K_{nm} , \qquad (12)$$

$$Q^K_m = - \int J^r_{im} \ddot{r}_{i0} \, dm . \qquad (13)$$

The symmetry properties should be especially noted.

3.3 GRAVITATIONAL FORCES

The same considerations about the kinematics of a generic mass element are applied to the contribution of gravity to the equations of motions. The corresponding terms may be generated by the principle of virtual work. This results in the following terms due to gravity (superscript G)

$$K^G_{mn} q_n = Q^G_m , \qquad (14)$$

where

$$K^G_{mn} = - g_i \int H^r_{imn} \, dm = K^G_{nm} \qquad (15)$$

and

$$Q^G_m = g_i \int J^r_{im} \, dm \qquad (16)$$

with i = 1, 2, 3.

3.4 AERODYNAMIC FORCES

For aerodynamic forces on the blade the well-known strip theory is applied. Only quasi-steady lift and drag forces acting on the aerodynamic center line are considered, but also the aerodynamic pitching moment can be included without difficulty. According to figure 3 the components of the aerodynamic force on a differential element of the blade are functions of the components of the (instanteous) local velocity

$$dF_i = dF_i(U_k) \qquad (17)$$

with i, k = 2, 3. These components can be written as

$$U_k = T_{lk}(w_l - \dot{a}_l) \qquad (18)$$

with l = 1, 2, 3, where the difference between the inertial components of oncoming flow velocity w_l and the inertial components of the kinematic velocity of the aerodynamic center $\dot{a}_l$ is transformed to the blade section system by T_{lk}. The inertial coordinates of the aerodynamic center are described similar to (3) as

$$a_i = a_{i0} + J^a_{in} q_n + \frac{1}{2} H^a_{ikn} q_k q_n + \dots \qquad (19)$$

and the elements of the transformation matrix are expanded as follows

$$T_{lk} = T_{lk0} + J^T_{lkn} q_n + \dots . \qquad (20)$$

The velocity and force components in the blade section system are expanded into Taylor series with respect to q_l and its time derivative. Now a model of aerodynamic force generation must be introduced. With (17) the nominal aerodynamic forces can be calculated using the nominal velocity components in the blade section system

$$dF_{i0} = dF_i(U_{k0}) . \qquad (21)$$

In addition, with

$$dF_{ik} = \frac{\partial dF_i}{\partial U_k} (U_{k0}) \qquad (22)$$

the derivative of (17) with respect to the velocity components is defined. The application of the virtual work principle yields after some manipulations the following aerodynamic contributions (superscript A)

$$D^A_{mn} \dot{q}_n + K^A_{mn} q_n = Q^A_m , \qquad (23)$$

where

$$D^A_{mn} = \int A_{im} \, dF_{ik} \, A_{kn} , \qquad (24)$$

$$K^A_{mn} = - \int dF_{i0} \, B_{imn} + A_{im} \, dF_{ik} \, C_{kn} , \qquad (25)$$

$$Q^A_m = \int dF_{i0} \, A_{im} . \qquad (26)$$

Integration has to be carried out over the blade span. The structure of the matrices is such that the force-generating terms are distinguishable from the kinematic terms, which are represented by the auxiliary matrices

$$A_{im} = T_{ji0} J^a_{jm} , \qquad (27)$$

$$B_{imn} = J^T_{jin} J^a_{jm} + T_{ji0} H^a_{jmn} , \qquad (28)$$

$$C_{kn} = J^T_{lkn}(w_l - \dot{a}_{l0}) - T_{lk0} \dot{J}^a_{ln} . \qquad (29)$$

3.5 ELASTIC FORCES

The use of generalized coordinates corresponding to eigenmodes of the blade simplifies the generation of linear elastic contributions to the equations of motion considerably. The elastic energy of the blade is simply

$$U = \frac{1}{2} K^E_{mn} q_m q_n \qquad (30)$$

with $m = n$ only, where the generalized stiffnesses are known from finite-element calculations or modal survey tests. Thus the derivation of the elastic contributions (superscript E) on the left-hand side

$$K^E_{mn} q_n \qquad (31)$$

is trivial in this case.

3.6 STRUCTURAL DAMPING FORCES

The same arguments as for the elastic forces apply also to the structural energy dissipation. Using modal damping for the elastic eigenmodes, one can write the dissipation function as

$$D = \frac{1}{2} D^{D}_{mn} \dot{q}_m \dot{q}_n \tag{32}$$

with $m = n$ only. The derivation results in the following structural damping contribution (superscript D) on the left-hand side

$$D^{D}_{mn} \dot{q}_n \tag{33}$$

for the individual blade.

3.7 MULTIBLADE COORDINATE TRANSFORMATION

So far the equations of motion of one, say the k-th, blade have been considered. They can be written as

$$\underline{\underline{M}}_k \ddot{\underline{q}}_k + \underline{\underline{D}}_k \dot{\underline{q}}_k + \underline{\underline{K}}_k \underline{q}_k = \underline{Q}_k \tag{34}$$

where the contributions of (7), (14), (23), (31), and (33) are collected at the appropriate side. It should be noted that (34) has periodic coefficients with a frequency as low as the rotational frequency of the rotor. It would be possible to generate equations for each blade using the appropriate azimuth angle and couple them afterwards at the hub with the nonrotating structure. But it is more advantageous to introduce so-called multiblade or rotor generalized coordinates instead, see /11/. The generalized coordinates of the individual blades are expressed as a sum of N terms

$$\underline{q}_k = \underline{q}_0 + \sum_{n=1}^{M} \left(\underline{q}_{nC} \cos n\,\varphi_k + \underline{q}_{nS} \sin n\,\varphi_k \right) + \underline{q}_d (-1)^{k-1} \tag{35}$$

where N represents the number of blades of the rotor. M is equal to (N-1)/2 if N is odd and (N-2)/2 if N is even. The last term occurs only in the latter case. In matrix form, (35) can be written as

$$\underline{q}_k = \underline{\underline{R}}\, \underline{q}_R \tag{36}$$

with a time-dependent transformation matrix $\underline{\underline{R}}$. For each blade (36) is introduced into (34), the result is multiplied from the left with $\underline{\underline{R}}$ transposed, and afterwards all matrices are summed up. The equations of motion appear as

$$\underline{\underline{M}}_R \ddot{\underline{q}}_R + \underline{\underline{D}}_R \dot{\underline{q}}_R + \underline{\underline{K}}_R \underline{q}_R = \underline{Q}_R \ . \tag{37}$$

Although the number of generalized coordinates increases compared to the single blade equations, the new form is more attractive because the lowest frequency in the matrix coefficients is increased from 1/rev to N/rev for rotors with more than two blades. So it is possible and justified to use a constant coefficient approximation of the resulting matrices. In some special cases like hovering of such a rotor, the periodicity in the coefficients vanishes altogether, even with anisotropic support conditions.

From manual derivation it can be recognized that many matrix elements simplify with the application of the multiblade coordinate transformation. But unfortunately, the necessary algebra is awkward because numerous trigonometric simplifications are involved. Rotor simulation programs can apply the multiblade coordinate transformation for each time step in numerical form. Using computer-algebra it should be possible to yield closed-form expressions, which are more efficient to compute.

4. COMPUTER PROGRAM

In the following a short description of the computer program written in REDUCE for the generation of literal expressions for the matrix elements is given.

4.1 GENERAL REMARKS

Parts of the program which are more or less subject to changes if a new model is generated are contained in "input files" discussed later. On the other hand, there are parts of the program which do not depend in any respect on the specific model under consideration. Five steps are assigned for each kind of force and are run successively. This program structure is shown in figure 4. The stepwise execution is advantagous, because it provides the option to view the intermediate results before going ahead with the next step. It should also be noted that the REDUCE output from one step can serve directly as input for the next. Renaming and reordering steps must possibly be run more times to find a desired output form, but usually this is not very time-consuming. If a more general model has been generated, it is possible, without rerunning the program from the beginning, to introduce certain simplifications at the last steps. This allows an efficient and consistent derivation of a series of simplified models from one master model. The contributions of forces of different physical origin are handled independently and it is assumed that they are summed up in the numeric program.

4.2 PROGRAM INPUT

Table 1 shows the input files, which must be specified (or modified) by the user for each step.

GLOBI serves for dimensioning purposes. Because the generalized coordinates and their time derivatives are handled internally as symbolic subscripted variables, they are declared "operators" which have an integer argument.

TIME declares the time dependency of the generalized coordinates q_n and the nominal state values of $\underline{q}_1$.

TRIGO contains a set of trigonometric "let"-rules to carry out possible simplifications.

WEIGHT introduces a weighting scheme, which will be used through the derivation process to neglect higher order terms consistently. "Weights" are assigned to parameters. The sum of weights of each term is checked at the beginning of the evaluation of an expression whether it exceeds a given "weight level." If this is the case the term will be deleted.

EVNOM is a small algebraic "procedure," which evaluates a symbolic expression using the nominal-state values of $\underline{q}_1$.

RNMOD performs the renaming of unsubscripted generalized coordinates to subscripted ones and introduces spatial modal functions and corresponding generalized coordinates for elastic displacements of the blade.

DEFMAT defines the matrices necessary to describe the kinematics of the blade.

MATREX combines the above-defined matrices in matrix expressions for inertial coordinates of a generic mass element and a generic element of the aerodynamic center line, respectively. Also, the coordinate transformation between the inertial system and the aerodynamic blade section system is established in matrix form. This is all done by using the matrix algebra facilities in REDUCE.

WIND defines the components of velocity of the oncoming wind in the inertial system as a column matrix.

GRAVI defines the components of acceleration due to gravity in the inertial system as a column matrix.

AERO describes the airload/velocity relationship in the blade section system.

DEFINT declares dependencies of the various parameters on the generic mass element and the element of the aerodynamic center line, respectively. "Linear operators" are introduced which carry out symbolically the integration over the blade volume and blade span. Thus integrals are left in an unspecified form and can be renamed by the user for convenience. In the subsequent numeric program, these integrals can be calculated for a given system once and for all. By this, spatial integration at each time step is avoided.

ROSUM defines the multiblade coordinate transformation matrix. Dependencies on blade index are declared. A "linear operator" with respect to the blade index is used to perform symbolic summation over all the blades using well-known rules for simplification.

RNSIM is used for giving new names to variables and parameters. Also posteriori simplifications can be introduced. Subexpressions can be replaced by auxiliary variables to reduce output length.

ORDER allows the user to change the internal order of variables and parameters. This affects the non-expanded form of an expression. So the user has a tool to create a well-structured output with multilevel bracketing. This is of vital importance for legibility and the efficiency of created FORTRAN code.

It should be noted that only a minimum amount of information about a specific model is necessary to run the program and no manual derivation beyond very basic modeling has to be carried out in advance.

4.3 PROGRAM OUTPUT

The results of the last step are usually given in three different forms. For visual inspection a list is printed on which the expressions are written with each term on a new line using a "natural" style with raised exponents. For input to later - perhaps interactive - REDUCE manipulations the results are stored on magnetic disk using single-line style. And finally FORTRAN-compatible statements are written on magnetic disk, which reflect the derived mathematical model. In the case of large expressions proper continuation cards and automatic decomposition in subexpressions are provided in REDUCE.

5. EXAMPLE

As an example, the suggested procedure is applied to the problem of a two-bladed wind turbine mounted on an elastic tower. The investigation of the aeroelastic stability of such systems has gained importance in recent years with the advent of large wind energy converters with high specific speeds which require very slender blades for optimum aerodynamic performance /12/.

5.1 SYSTEM DESCRIPTION

Figure 5 shows the system under consideration. Only the rotor equations are generated by computer algebra. A modal coupling procedure is employed in the numerical program to enforce compatibility and equilibrium at the interface of the subsystems rotor and tower. The rotor kinematics is described with three translations and three rotations of the hub (due to tower displacements), a teetering motion of the complete rotor, and bending deflections of the blades in flapwise and chordwise directions described by two representative coupled normal modes of the blade fixed at the pitch bearing. Two modes are necessary and sufficient to obtain literal expressions for all possible coupling elements in the desired matrices. The model degrees of freedom sum up to 11 linear degrees of freedom for the rotor. In addition the azimuth angle of the rotor and the collective pitch angle are treated without any linearization. Undersling and precone are provided. Teetering of the rotor can introduce differential pitch angles of the blades due to an arbitrarily large amount of pitch-flap coupling ($\tan \delta_3$). The blades can have an arbitrary amount of pretwist and are modeled as bodies with spatial extension perpendicular to the blade reference axis. The blades can bend elastically, but are assumed to be rigid in torsion. It is assumed that the first two coupled bending eigenmodes are sufficient to describe the elastic blade deflections in the investigation of the low-frequency behaviour of the coupled rotor/tower system. Foreshortening is accounted for in the derivation of the equations. Thus only eigenmodes of the nonrotating blades are necessary, which can also be determined by a modal survey test. Aerodynamic forces are modeled by two-dimensional quasi-steady theory for lift and drag on the blades. No dynamic inflow as described in /13/ or unsteady airfoil theory has been considered up to now. Only simple momentum theory is applied to calculate the mean velocity at the rotor disk. The weighting scheme employed is shown in table 2.

The tower is modeled by generalized coordinates corresponding to eigenmodes, which result from a finite-element calculation. The tower model includes a point mass at the rotor hub, representing the total rotor mass. Also the polar moment of inertia of the rotor was modeled at this point. Certainly, the same contributions of the rotor model to the rotor/tower system mass matrix must be canceled in order to include them only once.

5.2 STORAGE AND TIME REQUIREMENTS

Symbolic derivation of equations of motion of the system described above was executed using a virtual storage region of 1 MegaByte. The CPU times necessary for each step and each kind of force are summarized in table 3. The equations for the two nonlinear treated generalized coordinates are included. It can be seen that the largest amount of time is spent generating the auxiliary arrays, especially the Hessian. The largest requirement of CPU time for a single job was 3727 sec for inertial auxiliary arrays, followed by 2578 sec for a part of the aerodynamic auxiliary arrays. For the complete model CPU time amounts to approximately 3.5 hours. At first glance, this may be considered large compared to a single purely numeric generation of equations. But it should be kept in mind that the symbolic results can save considerable computer time when they are used to create a more efficient numeric computer program afterwards, which shall run many times.

5.3 RESULTS

In figure 6 some input, intermediate, and output expressions are shown, illustrating the generation of the (7,2) element of the rotor mass matrix (coupling between lateral motion of the hub and teetering of the rotor). The whole output amounts to about 800 statements on 1800 lines. A further compression of output can be achieved if common subexpressions of different matrix elements can be identified and set to global auxiliary variables. In the current REDUCE version auxiliary variables are introduced only for the current expression, so this must be done interactively using an editor. With "symbolic mode" procedures it should be possible to alleviate this problem, but up to now the use of this more elementary REDUCE was avoided.

Finally, figure 7 shows the deflections in an unstable Floquet solution which was computed numerically over one rotor revolution.

6. CONCLUDING REMARKS

A general purpose computer-algebra system has been applied to develop a program which generates literal equations of motion for rotary-wing aeroelastic problems. With this tool, it is possible to handle rather complex configurations without manual derivation and coding coefficients of the resulting algebraic expressions. Contrary to the numerical procedures, which are of course indispensable for general-purpose programs, the user has more insight into the structure of his mathematical model and the generated FORTRAN output allows for efficient numerical treatment afterwards. Considering the experience with problems of expression swell, the extent to which complexity of models can increase without excessive demands on the computer is not easily predictable. But several means have not been used until now, which may alleviate this problem in the future.

Various details of current modeling techniques for aeroelastic stability and dynamic response of rotary-wings should be incorporated, which have been omitted up to now. Linearized systems should be treated in the sense that the equilibrium reference state enters the equations. It must be noted that the described method is not limited in this respect. Also equations to obtain an approximate equilibrium reference state should be generated. In this context, the problem of geometric nonlinearities inherent in modern rotor design should be treated by a modal approach. From an aerodynamics point of view more accurate modeling of unsteady airloads is desirable. The first step towards this goal will be the use of dynamic inflow models in the derivation.

Although a modal approach was preferred in this paper, it can be easily imagined that symbolic element matrices for application of the finite element method may be derived along the same lines. Contrary to the modal approach, where integrals must be left unspecified in general, the analytical form of parameter variations and shape functions now permits closed-form expressions, thereby avoiding numerical integration over the element.

REFERENCES

1. Friedmann, P. P., "Formulation and Solution of Rotary-Wing Aeroelastic Problems," Proceedings of the International Symposium on Aeroelasticity, Nuremberg, Federal Republic of Germany, October 1981.

2. Young, M. I. and Lytwyn, R. T., "The Influence of Blade Flapping Restraint on the Dynamic Stability of Low Disk Loading Propeller-Rotors," Journal of the American Helicopter Society, Vol. 12, No. 4, October 1967, pp. 38-54.

3. Johnson, W., "Development of a Comprehensive Analysis for Rotorcraft," Part I: Rotor Model and Wake Analysis, Part II: Aircraft Model, Solution Procedure, and Applications, Vertica, Vol. 5, No. 3, 1981, pp. 99-129 and Vol. 5, No. 4., 1981, pp. 185-216.

4. Yun, D. Y. Y. and Stoutemyer, D., "Symbolic Mathematical Computation," Encyclopedia of Computer Science and Technology, Vol. 15 Supplement, ed. by J. Belzer, A. G. Holzman, and A. Kent, Marcel Dekker, 1980.

5. Kreuzer, E., "Symbolische Berechnung der Bewegungsgleichungen von Mehrkörpersystemen," Series: Schwingungstechnik - Laermbekämpfung, Fortschrittsberichte der VDI-Zeitschriften, Reihe 11, Nr. 32, 1979.

6. Nagabhushanam, J., Gaonkar, G. H., and Reddy, T. S. R., "Automatic Generation of Equations for Rotor-Body Systems with Dynamic Inflow for A Priori Ordering Schemes," Paper No. 37, Proceedings of the Seventh European Rotorcraft and Powered Lift Aircraft Forum, Garmisch-Partenkirchen, Federal Republic of Germany, September 1981.

7. Kießling, F. and Ludwig, D., "Berechnung der Eigenschwingungen von Rotorblättern mit der Methode der finiten Elemente," DFVLR-FB 81-07, March 1981.

8. Tobey, R. G. et al., "PL/I - FORMAC Symbolic Mathematics Interpreter," IBM Contributed Program Library, 360D-03.3.004, 1969.

9. Hearn, A. C., "REDUCE 2 User's Manual," University of Utah, Salt Lake City, UCP-19, 2nd Ed., 1973.

10. Kane, T. R. and Wang, C. F., "On the Derivation of Equations of Motion," J. Soc. Indust. Appl. Math., Vol. 13, No. 2, 1965, pp. 487-492.

11. Hohenemser, K. H. and Yin, S. K., "Some Applications of the Method of Multiblade Coordinates," Journal of the American Helicopter Society, Vol. 16, No. 2, April 1971, pp. 3-12.

12. Hütter, U., "Optimum Wind-Energy Conversion Systems," Ann. Rev. Fluid Mech., Vol. 9, 1977, pp. 399-419.

13. Ormiston, R. A., "Application of Simplified Inflow Models to Rotorcraft Dynamic Analysis," Journal of the American Helicopter Society, Vol. 21, No. 3, July 1976, pp. 34-37.

Weight level	2
Weights:	
Section coordinates	1
Undersling	1
Modal bending displacements	1
Modal axial displacements	2
Precone	1
Drag coefficient/lift slope	2

Table 2. Weighting scheme

Step	Computing time <sec>		
	Inertia	Gravity	Aerodynamics
1	3727	-	3125
2	1044	33	2288
3	495	22	550
4	140	15	176
5	136	14	177
Sum	5542	84	6316

Table 3. Computing times

File	Step				
	1	2	3	4	5
GLOBI	*	*	*	*	*
TIME	*	*	*		
TRIGO	*	*			
WEIGHT	*	*			
EVNOM	*				
RNMOD	*				
DEFMAT	*				
MATREX	*				
WIND	*				
GRAVI		*			
AERO		*			
DEFINT		*			
ROSUM			*		
RNSIM				*	
ORDER					*

Table 1. Input files

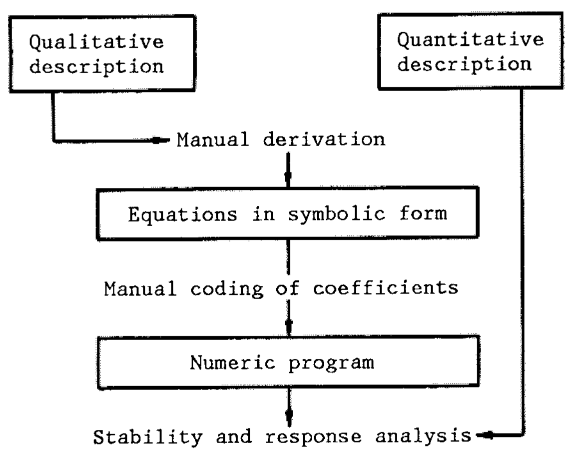

Figure 1a. Manual/numeric procedure

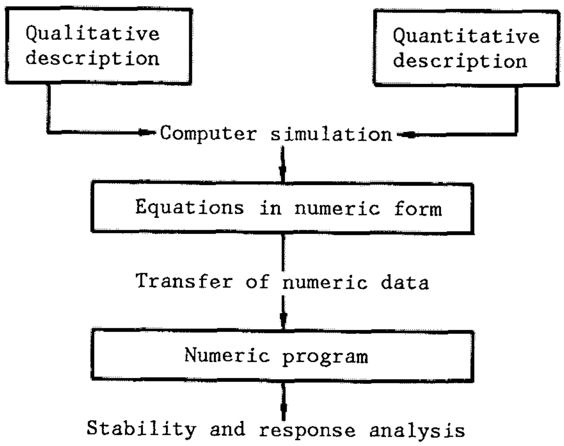

Figure 1b. Numeric procedure

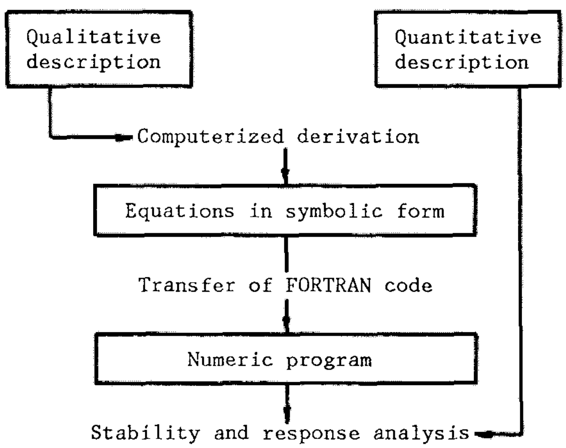

Figure 1c. Symbolic/numeric procedure

```
% EQUATION OF MOTION OF A PLANE PENDULUM

% LENGTH L, MASS M, ANGLE PHI WITH RESPECT TO
% THE VERTICAL X-AXIS (POSITIVE DOWNWARD);

% COORDINATES OF THE MASS;
X := L*COS(PHI)$ Y := L*SIN(PHI)$

% TRIGONOMETRIC SIMPLIFICATION RULE;
LET COS(PHI)**2 + SIN(PHI)**2 = 1;

% PHI DEPENDS ON T;
DEPEND PHI,T;

% FORM KINETIC ENERGY;
% USE DIFFERENTIATION OPERATOR DF;
KIN := M*(DF(X,T)**2 + DF(Y,T)**2)/2;

          2             2
KIN := (L *M*DF(PHI,T) )/2          <=== ANSWER

% FORM POTENTIAL ENERGY;
POT := - M*G*X;

POT :=  - G*L*M*COS(PHI)            <=== ANSWER

% ORDERING AND FACTORING;
ORDER G,M,L; OFF ALLFAC;

% LEFT-HAND SIDE OF LAGRANGE'S EQUATION;
DF(DF(KIN,DF(PHI,T)),T) + DF(POT,PHI);

                       2
G*M*L*SIN(PHI) + M*L *DF(PHI,T,2)   <=== ANSWER

END;
```

Figure 2. Sample REDUCE program

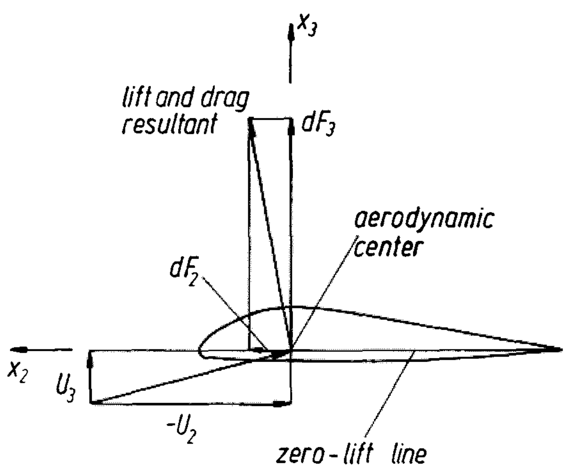

Figure 3. Velocities and aerodynamic forces at a blade section

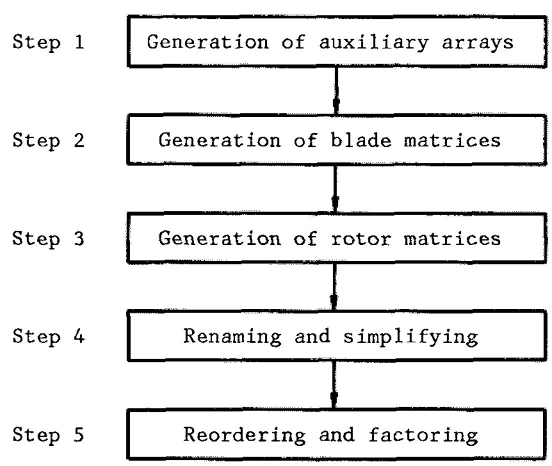

Figure 4. Program structure

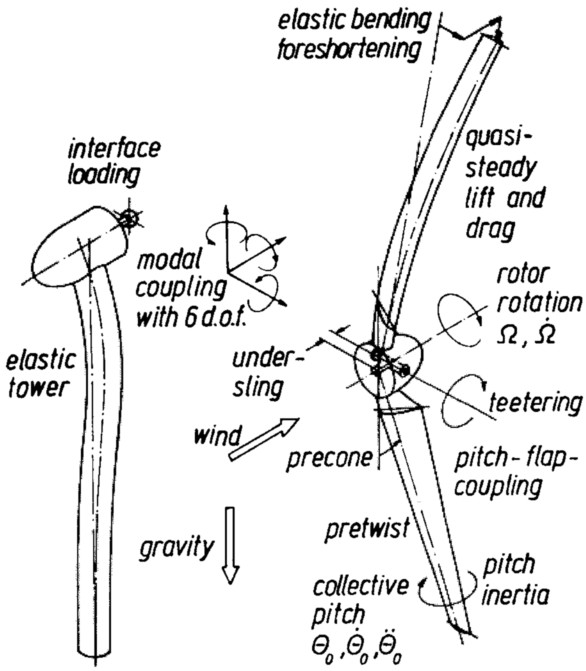

Figure 5. Wind turbine model

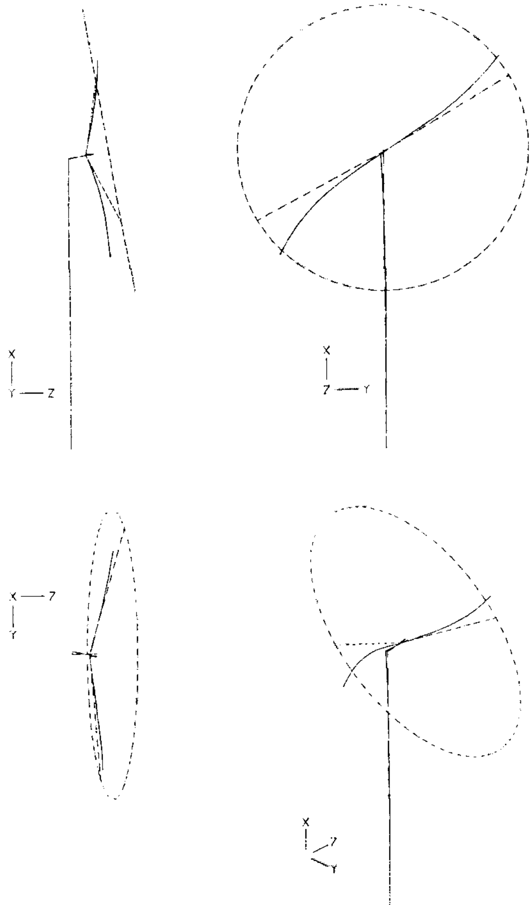

Figure 7. Time step of Floquet solution

Input (MATREX) : Inertial coordinates of a generic element of mass; RN, RA, RX, RK, RQ are (3,1) column matrices and PSN, TEN, PHN, PHK, BEK, BEA, DTK, TET are (3,3) elementary rotational transformation matrices defined in input file DEFMAT.

```
MATRIX RI (3,1)$

RI := RN + PSN*TEN*PHN*PHK*BEK*(RA + BEA*(RX + DTK*TET*(RK + RQ)));
```

Step 1 result: elements of the Jacobian of RI evaluated at q = 0

```
JLRI(1,2) := 0$ JLRI(2,2) := 1$ JLRI(3,2) := 0$

JLRI(2,7) :=  - X*SIN(QN(1))*BETA + Y*COS(QN(2))*TDEL*SIN(QN(1))*BETA +
Y*SIN(QN(2))*TDEL*COS(QN(1)) - Y*SIN(QN(2))*SIN(QN(1)) - Z*SIN(QN(2))*TDEL*SIN(QN(1))*BETA +
Z*COS(QN(2))*TDEL*COS(QN(1)) - Z*COS(QN(2))*SIN(QN(1)) - SIN(QN(1))*ZA$
```

Step 2 result: element of the blade mass matrix

```
BLKM(7,2) :=  - BETA*TDEL*SIN(QN(2))*SIN(QN(1))*IZ + BETA*TDEL*COS(QN(2))*SIN(QN(1))*IY -
BETA*SIN(QN(1))*IX - ZA*SIN(QN(1))*I1 + TDEL *SIN(QN(2))*COS(QN(1))*IY +
TDEL*COS(QN(2))*COS(QN(1))*IZ - SIN(QN( 2))*SIN(QN(1))*IY - COS(QN(2))*SIN(QN(1))*IZ$
```

Step 3 result: element of the rotor mass matrix

```
RLKM(7,2) := 2*(TDEL*COS(QN(2))*IZ*COS(QN(1)) + TDEL*COS(QN(2))*IY*BETA*SIN(QN(1)) -
TDEL*SIN(QN(2))*IZ*BETA*SIN(QN(1)) + TDEL*SIN( QN(2))*IY*COS(QN(1)) - COS(QN(2))*IZ*
SIN(QN(1)) - SIN(QN(2))*IY* SIN(QN(1)) - ZA*I1*SIN( QN(1)) - IX*BETA*SIN(QN(1)))$
```

Step 4 result: element of the rotor mass matrix (renamed variables)

```
RLKM(7,2) := 2*CP*TDEL*CT*IZ + 2*CP*TDEL*ST*IY + 2*SP*TDEL*BETA*CT*IY -
2*SP*TDEL*BETA*ST*IZ -
2*SP*BETA*IX - 2*SP*ZA*I1 - 2*SP*CT*IZ - 2*SP*ST*IY$
```

Step 5 result: element of the rotor mass matrix (factored form)

```
RLKM(7,2) := 2*(CP*TDEL*(CT*IZ + ST*IY) + SP*(TDEL*BETA*(CT*IY - ST*IZ) - BETA*IX - ZA*I1 -
CT*IZ - ST*IY))$
```

Step 5 result: element of the rotor mass matrix (FORTRAN code)

```
1234567 <=== Card column

      RLKM(7,2)=2.*(CP*TDEL*(CT*IZ+ST*IY)+SP*(TDEL*BETA*(CT*IY-
     . ST*IZ)-BETA*IX-ZA*I1-CT*IZ-ST*IY))
```

Figure 6. REDUCE results