Journal of Sound and Vibration (1986) 109(1), 65-78

SYMBOLIC COMPUTING AS A TOOL IN WIND TURBINE DYNAMICS

A. D. GARRAD[†] AND D. C. QUARTON

Wind Energy Group, Taylor Woodrow Construction, 345 *Ruislip Road, Southall UB*1 2*QX, England*

(*Received* 26 *October* 1984, *and in revised form* 25 *September* 1985)

The problem of deriving equations of motion that describe the coupled tower-rotor system of a wind turbine is discussed. The mathematical formulation of these equations is described first in a form suitable for manual derivation and then as a step by step process suitable for automation. Reasons for and experience in using a symbolic computing system to undertake this work for large models are described. The general approach is illustrated by means of a very simple model. Finally, some results of the stability analysis of the simple example and an eleven degree of freedom three-bladed model are presented together with a set of measured and predicted blade loads for a 250 kW wind turbine.

1. INTRODUCTION

We recently encountered a problem of algebraic manipulation in wind turbine dynamics which was conceptually very straightforward but immensely tedious to perform. We required a solution to this problem and were interested in finding a means of performing it automatically, first as a check on the manual calculation, and then as a method in its own right. The power and possible application of computers as "equation crunchers" rather than "number crunchers" was a revelation to us. In this paper we shall not seek to break any new ground, but rather describe the analysis which we conducted as an illustration of the possible applications of symbolic or algebraic computing as a tool in engineering analysis. Our particular problem was concerned with formulating the equations of motion of a wind turbine system. This problem will be used here as an example, but it will become obvious that many problems in engineering dynamics are of a similar nature.

Symbolic computing systems have been in existence for some 20 years. References were made to this application of digital computers as early as 1953 [1, 2]. The systems seem to have been developed for high energy physics applications and, indeed, the system used by us (REDUCE) bears definite signs of its origins in this respect. Their application to engineering in general and engineering mechanics in particular seems to have been very limited. Some work on helicopter dynamics has been reported by Nagabhushanam *et al.* [3]; a review of various systems applied to structural dynamics has been given by Noor and Anderson [4]. A layman's guide to computer algebra and its applications was given by Pavelle *et al.* [5].

There are quite a number of systems available, notably FORMAC, MACSYMA, REDUCE, SCRATCHPAD and most recently SMP. Some of these were reviewed by Jensen and Niordsen [6]. The present authors have first hand experience only of REDUCE and our limited knowledge of symbolic computing indicates that this system is the most widely used in engineering applications. Some recent papers have appeared in which

[†] Present address Garrad Hassan & Partners, 10 Northampton Square, London EC1M 5PQ, England.

problems very similar to our own have been treated. Kiessling [7, 8] discussed REDUCE in the context of wind turbine problems, but unfortunately both papers appeared in rather obscure publications. Reference [8] is a very thorough description of the use of symbolic computing in this type of work. Koppens [9] has considered some rotorcraft problems, less complex than those considered by Kiessling or in the present paper, but he retained the non-linear terms which, of course, is no hardship when using the computer to perform the algebra.

The growing interest in symbolic computing is aptly demonstrated by the advent of a new technical journal: *Journal of Symbolic Computation.* In a paper published there Fitch [10] has summed up its position in the engineering field: "(We)... undertook a programme for the education of... companies in both Sweden and Britain in computer algebra, and we discovered that once these techniques had been shown to them they immediately recognized that they would be able to ask questions that they had been suppressing as incapable of solution. Many of these companies are now acquiring algebra systems. At present it is unfortunately still the case that much algebra is being done by hand". That is the view of a computer scientist. We hope that the present paper will reinforce his view and encourage many other engineers to use this powerful aid to analysis.

2. WIND TURBINE DYNAMICS

The main problem in producing a mathematical model of a wind turbine is the fact that the rotor exercises gross angular movements with respect to the support structure. In standard structural analysis techniques, such as finite element packages, it is assumed that the structure has a mean position about which deflections, albeit large deflections, occur. This is evidently not the case for a wind turbine. The analyst is therefore forced to develop the equations of motion from first principles. The goal of the analysis is to be able to predict the stability of the system and to be able to calculate the forced response.

3. DERIVATION OF THE EQUATIONS OF MOTION

3.1. DEFINITION OF THE CO-ORDINATE SYSTEMS

It is common practice when faced with the derivation of the equations of motion of a complex dynamic system to use the well-known approach due to Lagrange. This requires the calculation of an expression for the various forms of energy in the system, the kinetic energy T, the strain energy U and the dissipation energy U_D. After having calculated these quantities the Lagrangian may be derived and equated with an expression for the generalized force Q.

For a system such as we are considering the computation of the strain and dissipation energies is relatively straightforward compared with the kinetic energy and generalized force.

For the wind turbine problems we considered standard methods were used to set up the description of an arbitrary point on one of the blades. A series of transformations (rotations and translations) were calculated that allowed the position vector of an arbitrary point on the blades to be expressed in inertially fixed axes. These transformations are listed below and sketched in Figure 1.

The inertially fixed co-ordinate system is defined such that it is aligned along the unperturbed tower. The remaining co-ordinate systems are defined as follows: $\Gamma_0(x_0, y_0, z_0)$ is the inertially fixed system; $\Gamma_1(x_1, y_1, z_1)$ is obtained by translation of Γ_0 by the tower head displacement vector, $\mathbf{n}_1$; $\Gamma_2(x_2, y_2, z_2)$ is obtained by torsional rotation, θ_y, about y_1; $\Gamma_3(x_3, y_3, z_3)$ is obtained by "nodding" rotation, θ_x, about x_2; $\Gamma_4(x_4, y_4, z_4)$ is

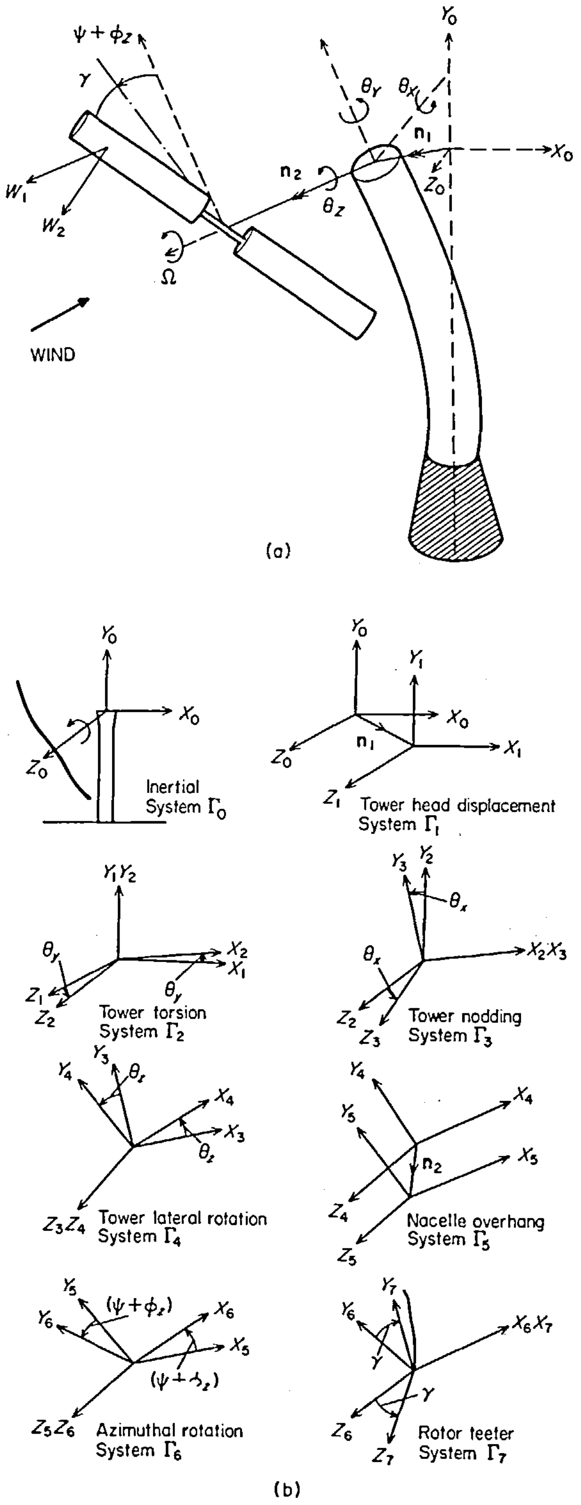

Figure 1. (a) System degrees of freedom; $\mathbf{n}_1$, tower head displacement; θ_y, tower torsion; θ_x, tower nodding; θ_z, tower lateral rotation; $\mathbf{n}_2$, nacelle overhang; $\psi+\phi_z$, azimuthal rotation; γ, rotor teeter; W_1, rotor flapwise bending; W_2, rotor edgewise bending; (b) co-ordinate systems.

obtained by lateral rotation, θ_z, about z_3; $\Gamma_5(x_5, y_5, z_5)$ is obtained by translation of Γ_4 by the overhang vector $\mathbf{n}_2$; $\Gamma_6(x_6, y_6, z_6)$ is obtained by azimuthal rotation, $(\psi+\phi_z)$, about z_5; $\Gamma_7(x_7, y_7, z_7)$ is obtained by teeter rotation, γ, about x_6.

The azimuthal rotation, $(\psi+\phi_z)$ is the sum of the steady state and power train perturbation components.

The rotational matrices defining the co-ordinate transformations are standard and may be easily determined from the definition of the co-ordinate systems given above.

Generalized co-ordinates were used that corresponded to the following physical degrees of freedom; rotor teeter; rotor azimuth and power train rotation; flapwise and edgewise displacement including arbitrarily large initial deflections; tower torsion; tower fore-aft and side-to-side displacement.

3.2. DERIVATION OF KINETIC ENERGY

The kinetic energy of the wind turbine structure may be written as

$$T=\tfrac{1}{2}\int_{\text{rotor}} \mathbf{V}_6^{\mathrm{T}} M_R \mathbf{V}_6 \,\mathrm{d}r+\tfrac{1}{2}\int_{\text{tower}} \mathbf{V}_{0_T}^{\mathrm{T}}[M_T]\mathbf{V}_{0_T}\,\mathrm{d}h, \tag{1}$$

where $\mathbf{V}_{0_T}$ and $\mathbf{V}_6$ are the absolute velocity vectors of a point on the tower and on the rotor, expressed in Γ_0 and Γ_6 co-ordinate systems respectively. The tower distributed mass and inertia properties are contained in the matrix M_T.

After considerable manipulation the velocity of a point on the rotor can be expressed as

$$\begin{aligned}\mathbf{V}_6=&[T_{\bar{\psi}}]^{\mathrm{T}}[T_{\theta_z}]^{\mathrm{T}}[T_{\theta_x}]^{\mathrm{T}}[T_{\theta_y}]^{\mathrm{T}}\dot{\mathbf{n}}_1+[T_{\bar{\psi}}]^{\mathrm{T}}[T_{\theta_z}]^{\mathrm{T}}[T_{\theta_x}]^{\mathrm{T}}(\dot{\boldsymbol{\theta}}_y\times([T_\gamma]\mathbf{R}_7+[T_{\theta_x}][T_{\theta_z}]\mathbf{n}_2))\\&+[T_{\bar{\psi}}]^{\mathrm{T}}[T_{\theta_z}]^{\mathrm{T}}(\dot{\boldsymbol{\theta}}_x\times([T_\gamma]\mathbf{R}_7+[T_{\theta_z}]\mathbf{n}_2))+[T_{\bar{\psi}}]^{\mathrm{T}}[\dot{\boldsymbol{\theta}}_z\times([T_\gamma]\mathbf{R}_7+\mathbf{n}_2))\\&+\dot{\bar{\boldsymbol{\psi}}}\times[T_\gamma]\mathbf{R}_7+[T_\gamma](\dot{\mathbf{R}}_7+\dot{\boldsymbol{\gamma}}\times\mathbf{R}_7),\end{aligned} \tag{2}$$

where the rotational matrices $[T]$ are standard and may easily be determined from the definition of the co-ordinate systems.

After the rotor absolute velocity has been calculated in this way, the kinetic energy is derived by substitution of the generalized co-ordinates into equation (1). The resultant expression is not included in this text since it is extremely cumbersome and of no great physical significance.

3.3. DERIVATION OF THE GENERALIZED FORCE

The generalized force corresponding to the arbitrary generalized co-ordinate q is given by,

$$Q=\int \mathbf{F}_0\cdot\frac{\partial\mathbf{R}_0}{\partial q}\,\mathrm{d}r, \tag{3}$$

where $\mathbf{F}_0$ is the aerodynamic force vector acting at a point on the rotor defined by the displacement vector $\mathbf{R}_0$.

The rotor-fixed force vector $\mathbf{F}_7$ is more convenient to work with and is derived from consideration of the local aerodynamic lift and drag loads resolved into the blade edgewise and flapwise directions and is given by

$$\mathrm{d}\mathbf{F}_7=\begin{pmatrix}\tfrac{1}{2}\rho Uc(C_L U_{P_7}-C_D U_{T_7})\,\mathrm{d}r\\ \tfrac{1}{2}\rho Uc(C_L U_{T_7}+C_D U_{P_7})\,\mathrm{d}r\end{pmatrix}, \tag{4}$$

where U_{T_7} and U_{P_7} are the edgewise and flapwise components of the relative wind velocity vector $\mathbf{V}_{\mathrm{Rel}_7}$ expressed in Γ_7 rotor-fixed axes, and $U=(U_{T_7}^2+U_{P_7}^2)^{1/2}$.

The relative wind velocity vector defines the velocity of the wind with respect to the structural velocity of the rotor and is computed from

$$\mathbf{V}_{\mathrm{Rel}_7}=[T_\gamma][T_{\bar{\psi}}][T_{\theta_z}][T_{\theta_x}][T_{\theta_y}](\mathbf{V}_{w_0}-\dot{\mathbf{R}}_0), \tag{5}$$

where $\mathbf{V}_{w_0}$ is the absolute velocity of the wind expressed in inertial axes and the $\dot{\mathbf{R}}_0$ is the structural velocity.

The aerodynamic force vector $\mathbf{F}_0$ is derived by transforming $\mathbf{F}_7$. The generalized force may then be computed from equation (2) after having first substituted into it the generalized co-ordinates.

3.4. EQUATIONS FOR THREE-BLADED MACHINES

It is interesting to consider the slightly different but strongly related problem of a three-bladed rotor. The discussion above described the derivation of the equations of motion of a two-bladed rotor starting with an arbitrary point on the blade. The teeter freedom requires the two-bladed rotor to be treated as a single entity rather than as two independent blades. In the basic steps involved an arbitrary point was considered and it is only the specification of the mode shapes themselves that narrows the analysis to a two-bladed case. One may, therefore, take the kinetic energy derivation of a blade on a flexible tower as a starting point for the analysis of a three-bladed machine. It was mentioned above that two-bladed rotors pose considerable numerical problems in stability calculations. A three-bladed rotor is much simpler since the co-ordinates may be transformed to remove the periodic terms. Dugundji and Wendell [11] have given a good review of this aspect of wind turbine dynamics. In our case we were able to make use of the single-blade analysis that we had derived for the two-bladed work. Like the derivation of the equations of motion themselves, the transformation from physical co-ordinates into multi-blade co-ordinates can be very simply specified and was therefore suitable for symbolic computing.

To illustrate this transformation consider a state vector that describes a three-bladed rotor supported by a flexible tower. The blades are modelled in terms of their first flapping and lead-lag modes. $q_{1,i}$ denotes the generalized co-ordinate of the first flapping mode of blade i and $p_{1,i}$ the first lead-lag co-ordinate.

ϕ_z is the power train angular deflection, n, s, θ_y are the fore-aft, side-to-side and torsional generalized co-ordinates of the tower. The state vector of the system is $\mathbf{x}$ where

$$\mathbf{x}^{\mathrm{T}}=[q_{1,1} \quad q_{1,2} \quad q_{1,3} \quad p_{1,1} \quad p_{1,2} \quad p_{1,3} \quad \phi_z \quad n \quad s \quad \theta_y].$$

The system can be transformed into multi-blade co-ordinates by making standard substitutions and defining a new state vector $\mathbf{x}^1$:

$$\mathbf{x}^{1\mathrm{T}}=[q_0 \quad q_s \quad q_c \quad p_0 \quad p_s \quad p_c \quad \phi_z \quad n \quad s \quad \theta_y].$$

If the untransformed equation of motion was

$$[M]\ddot{\mathbf{x}}+[C]\dot{\mathbf{x}}+[K]\mathbf{x}=\mathbf{0},$$

where $[M]$, $[C]$ and $[K]$ are periodic, then the new equation of motion is

$$[M^1]\ddot{\mathbf{x}}^1+[C^1]\dot{\mathbf{x}}^1+[K^1]\mathbf{x}^1=\mathbf{0},$$

where

$$[M^1]=[P]^{\mathrm{T}}[M][P], \qquad [C^1]=[P]^{\mathrm{T}}(2[M][\dot{P}]+[C][P]),$$
$$[K^1]=[P]^{\mathrm{T}}([M][\ddot{P}]+[C][\dot{P}]+[K][P]),$$

which contain only constant terms due to the resulting trigonometric summations. Here

$$[P]=\begin{bmatrix}[Q] & 0 & 0\\ 0 & [Q] & 0\\ 0 & 0 & [I]\end{bmatrix}, \qquad [Q]=\begin{bmatrix}1 & \sin\psi_1 & \cos\psi_1\\ 1 & \sin\psi_2 & \cos\psi_2\\ 1 & \sin\psi_3 & \cos\psi_3\end{bmatrix},$$

Where $[I]$ is the 4×4 identity matrix and the variable ψ_i is the azimuth of the ith blade.

3.5. SYSTEMATIC DERIVATION OF THE LAGRANGIAN AND GENERALIZED FORCE

Equation (1) appears fairly innocent, but reference to equation (2) which shows the algebraic form of $\mathbf{V}_6$ demonstrates that calculation of T_R is formidable indeed. The complexity and tedium of the algebra involved becomes still worse when the derivation of the equations of motion is considered and the evaluation of the Lagrangian is attempted. The approach adopted to derive the expression for T_R is fairly standard and follows very closely that described by Ottens and Zwaan [12]. One attempts to minimize the algebraic manipulation by avoiding the calculation of the position vector and then taking its time derivative to calculate the velocity which would be the most direct method. The complexity is reduced by calculating the inertial velocity in an intermediate set of axes, the result being $\mathbf{V}_6$. Nevertheless, the algebra involved is extremely tedious and, in the view of the present authors, approaches the limits of human endurance or at least reliability. This sort of manipulation can, of course, be performed, but examination of the resulting expressions, which must be checked, suggested that short of performing a wholly independent analysis and comparing the results there was no possible, reliable method of checking.

The derivation of the kinetic energy and the steps involved in the derivation of the equations of motion are very well defined. It is possible to outline a general scheme for calculation of the Lagrangian in a step by step method: (1) specify the position vector ($\mathbf{R}_7$) of an arbitrary point of the blade in blade axes; (2) specify the series of transformations that are required to express the position vector in inertial axes, $\mathbf{R}_0=[T]\mathbf{R}_7$; (3) take the derivative of $\mathbf{R}_0$ with respect to time in order to calculate the inertial velocity, $\dot{\mathbf{R}}_0$; (4) calculate the kinetic energy of the point mass and integrate over the rotor, $\int m\dot{r}_i^2\,\mathrm{d}r=T_R$; (5) form the Lagrangian for each of the generalized co-ordinates.

By following the five steps specified above it is possible to calculate the equation of motion for each of the generalized co-ordinates. Thus, after having completed step 4, step 5 may be repeated for each of the co-ordinates so that the complete system equations are produced. In the list above a very inelegant method has been adopted to calculate the velocity—the position vector is simply differentiated with respect to time. This approach makes the algebra still more tedious but makes the specification of the steps considerably simpler. Faced with the problem of checking the derivation of the equations of motion, and realizing the mechanical way in which they were derived, suggested to us that our problem was ideally suited to solution by means of an algebraic computing system.

As with the Lagrangian, the derivation of the generalized force vector appears to be a straightforward mathematical procedure. However, the algebraic complexity of the derivation is again a great obstacle to a manual solution. The problem is therefore well-suited to the application of symbolic computing techniques. It is possible to define a series of mathematical steps for calculation of the generalized force: (1) specify the wind velocity vector $\mathbf{V}_{w_0}$ in inertial axes and subtract the inertial rotor velocity, $\dot{\mathbf{R}}_0$; (2) compute the relative wind velocity in blade axes by means of the rotational transformations, $\mathbf{V}_{\mathrm{Rel}_7}=[T]^{\mathrm{T}}(\mathbf{V}_{w_0}-\dot{\mathbf{R}}_0)$ (see equation (5));(3) compute the aerodynamic force vector in blade axes by means of $\mathbf{V}_{\mathrm{Rel}_7}$ and the lift and drag coefficients; (4) express the aerodynamic force vector in inertial axes, $\mathbf{F}_0=[T]\mathbf{F}_7$; (5) take the partial derivative of $\mathbf{R}_0$ with respect to the generalized co-ordinate to give the vector, $\partial\mathbf{R}_0/\partial q$; (6) form the dot product of $\mathbf{F}_0$ with $\partial\mathbf{R}_0/\partial q$ and integrate over the rotor to give the generalized force.

The algebra involved in the derivation of the generalized force vector is clearly complex and yet the mathematical procedure remains relatively straightforward. It is true to say, however, that the procedure requires a greater degree of careful consideration at each step compared to the easily automated derivation of the Lagrangian.

4. SOME EXAMPLES OF SYMBOLIC MANIPULATIONS

Symbolic computing comes into its own when tackling complicated and tedious problems. The type of problem for which we have found it useful is described in section 3. It will come as no surprise to workers in the field of dynamics that the matrices involved in these problems are very large and, although it is vital that they are correct, it is difficult, if not impossible, to glean any physical understanding of the system they describe by scrutinizing them term by term.

The problems described above have been analyzed by using symbolic computing techniques but, rather than tackle a large problem in detail here, which will result in large and complex algebraic expressions, it is more illustrative to consider a very simple example which may be followed through from beginning to end.

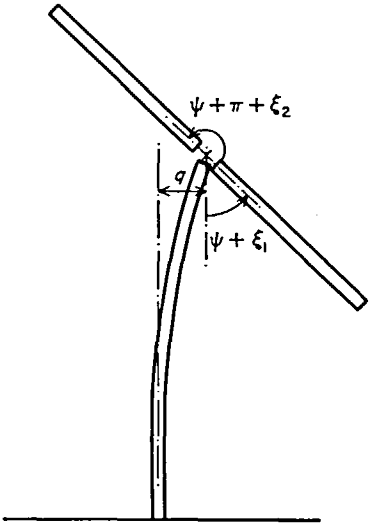

Figure 2. Three degree of freedom model. $\psi=\Omega t$.

Consider the wind turbine model shown in Figure 2. This is a simple three degree of freedom model first described by Kaza *et al.* [13]. It bears little physical resemblance to a real wind turbine, in fact its dynamic characteristics correspond rather more closely to a helicopter; however, it does do well as an illustrative model. The degrees of freedom contained in the model are tower head lateral motion and blade lead-lag motion. The derivation of the equations of motion of this system follows exactly the same steps as for the larger system described in section 3. The equivalent REDUCE steps, together with the inertial parts of the equations themselves, are given in Table 1.

TABLE 1

A simple example of REDUCE

```
% The following REDUCE program calculates the Lagrangians of
% the simple 3 d.o.f. system described in Figure 2.
%
MATRIX MPSI(3,3),VTOW(3,1),VRI(3,1),VR2(3,1),VR3(3,1),VR3DOT(3,1)$
%
% Algebraic procedure for differentiation w.r.t. time:
%
PROCEDURE DIFFRT(A)$
BEGIN
RETURN QDOT*DF(A,Q)
    +PSIDOT*DF(A,PSI)$
END$
%
% Algebraic procedure for determination of a Lagrangian:
%
PROCEDURE LAG(B,C)$
BEGIN
      DD:=DF(TT,B)$
      AA:=DF(TT,C)$
RETURN OMEGA*DF(AA,PSIØ)
    +QDOT*DF(AA,Q)
    +QDOTDOT*DF(AA,QDOT)
    +Z1DOT*DF(AA,Z1)
    +Z1DOTDOT*DF(AA,Z1DOT)
    +Z2DOT*DF(AA,Z2)
    +Z2DOTDOT*DF(AA,ZDOT)
    -DD$
END$
%
% MPSI is the azimuthal rotation matrix:
%
MPSI:=MAT((COS(PSI),-SIN(PSI),Ø),(SIN(PSI),COS(PSI),Ø),(Ø,Ø,1))$
%
% VTOW is the tower head displacement defined in inertial axes:
%
VTOW:=MAT((Q),(Ø),(Ø))$
%
% VR1 is the position vector of an arbitrary point on the blade in blade axes:
%
VR1:=MAT((Ø),(RR),(Ø))$
VR2:=MPSI*VR1$
VR3:=VR2+VTOW$
CLEAR MPSI,VTOW,VR1,VR2$
%
% VR3DOT is the general blade velocity vector defined in inertial axes:
%
VR3DOT:=MAT((DIFFRT(VR3(1,1))),(DIFFRT(VR3(2,1))),(DIFFRT(VR3(3,1))))$
CLEAR VR3$
%
% VSQ is the velocity squared expression for an arbitrary blade station:
%
VSQ:=VR3DOT(1,1)*VR3DOT(1,1)+VR3DOT(2,1)*VR3DOT(2,1)+
      VR3DOT(3,1)*VR3DOT(3,1)$
LET SIN(PSI)=SINPSI$
LET COS(PSI)=COSPSI$
LET COSPSI**2=1-SINPSI**2$
VSQ:=VSQ$
```

TABLE 1 (*cont.*)

```
LET SINPSI**2=1-COSPSI**2$
VSQ:=VSQ$
%  Z1 and Z2 are the perturbation co-ordinates for blades 1 and 2:
%
PSIDOT:=OMEGA+Z1DOT$
LET SIN(Z1)=Z1$
LET COS(Z1)=1-Z1**2/2$
SINPSI:=SIN(PSIØ)*COS(Z1)+COS(PSIØ)*SIN(Z1)$
COSPSI:=COS(PSIØ)*COS(Z1)-SIN(PSIØ)*SIN(Z1)$
%
%  VSQ1 and VSQ2 are the velocity squared expressions for blades 1 and 2:
%
VSQ1:=VSQ$
VSQ2:=SUB(Z1=Z2,Z1DOT=Z2DOT,SIN(PSIØ)=-SIN(PSIØ,COS(PSIØ)=
      -COS(PSIØ),VSQ1)$
VSQTOT:=VSQ1+VSQ2$
CLEAR VSQ,VSQ1,VSQ2$
%
%  TT is the total kinetic energy of the rotor:
%
TT:=M*VSQTOT/2$
%
%  Definition of blade inertia IB and first mass moment integral SB:
%
LET M*RR*RR=IB$
TT:=TT$
LET M*RR=SB$
TT:=TT$
CLEAR Q,QDOT,Z1,Z1DOT,Z2,Z2DOT$
%
%  The Lagrangians are now computed:
%
LAGQ:=LAG(Q,QDOT)$
LAGZ1:=LAG(Z1,Z1DOT)$
LAGZ2:=LAG(Z2,Z2DOT)$
%
%  A weight level is defined for linearization of the Lagrangians.
%  The linearized expressions are then output;
%
WTLEVEL 1$
WEIGHT Q=1,QDOT=1,QDOTDOT=1,Z1=1,Z1DOT=1,Z1DOTDOT=
      1,Z2=1,Z2DOT=1,Z2DOTDOT=1$
LAGQ:=LAGQ;
LAGZ1:=LAGZ1;
LAGZ2:=LAGZ2;
END$
%  Lagrangian for 'q' generalized coordinate;
                                                        2
LAGQ:=COS(PSIØ)*Z2DOTDOT*SB-COS(PSIØ)*Z2*OMEGA *SB
      -COS(PSIØ)
                                          2
    *Z1DOTDOT*SB+COS(PSIØ)*Z1*OMEGA *SB-2*SIN(PSIØ)*
    Z2DOT*OMEGA*SB+2*SIN(PSIØ)*Z1DOT*OMEGA*SB+2*M*QDOTDOT
%  Lagrangian for 'Z1' generalized coordinate;
LAGZ1:=-COS(PSIØ)*QDOTDOT*SB+Z1DOTDOT*IB
%  Lagrangian for 'Z2' generalized coordinate:
LAGZ2:=COS(PSIØ)*QDOTDOT*SB+Z2DOTDOT*IB
```

The steps described in the table are fairly straightforward for any reader familiar with FORTRAN and dynamics. Some general comments may, however, be made. REDUCE has the capability of implementing "Procedures" that are very much akin to FORTRAN subroutines. In the example here we have used two procedures, one for differentiation with respect to time and one to form the Lagrangian as defined in equation (1). Note that REDUCE can only perform partial differentiation. The derivation of the equations of motion involves considerable manipulation of matrices. The simple example cited here shows how REDUCE can perform these operations. It is often necessary, even when manipulation is being performed automatically, to linearize expressions. This may simply be done by REDUCE by using the "WEIGHT" statement. This statement is used at the end of the example to assign weights to the individual variables and, given a general weight level of 1, all quadratic and higher order terms are discarded. The resulting Lagrangians for each of the three degrees of freedom are then calculated and printed at the end of the example.

The WEIGHT command may also be used to reduce the size of the kinetic energy expression. When considering a particular generalized co-ordinate many of the terms are redundant since they disappear when the expression is differentiated. Judicious use of the WEIGHT and WTLEVEL statements avoids the computation of the redundant terms which brings about significant reductions in the size of the expressions used. Failure to carry out this procedure does, in our experience, lead to unacceptably large expressions which brings the calculation to a halt when storage is exhausted. This problem of mid-calculation "expression swell" is very common and has been discussed by Nagabhushanam *et al.* [3].

One further, and very useful, facility that REDUCE possesses is the ability to write expressions to files in FORTRAN format thus enabling the computer to both derive the equation of motion and transcribe them into a FORTRAN program for numerical solution. This final step helps to eliminate further possible sources of careless mistakes, since a manual transcription process is a very likely place for such problems to arise.

5. SOME COMMENTS ABOUT THE USE OF REDUCE

The automatic derivation of the equations of motion of large dynamical systems brings with it two main advantages: increased confidence in the results and increased scope in the scale of problems that may be tackled. Some limitations are still imposed either by financial constraints or limitations in computer storage capacity, but, in principle, the size and complexity of a mathematical model need not be limited by the size and complexity of the algebraic expressions which it produces.

The analyses described here exercized only a small part of the REDUCE system. We have used matrix manipulation and differential calculus. The system contains a great many more analytical facilities particularly in the field of high energy physics. For the engineering user the integration and factorization facilities are probably the most relevant. It is interesting to note that the integrator does not simply run through a "look-up table" of known integrals but tackles each one on a rigorous and systematic basis. There is still some considerable work to be done on this aspect of the system; for example, integrals containing square roots present great difficulties and REDUCE often fails to produce an answer. In view of the indirect nature of integration this is not surprising. Much work is evidently underway on this subject at present.

We have found REDUCE very useful. We have also found it to be difficult to implement and sometimes very frustrating to use. It would be erroneous to give the impression that problems such as those described in this paper have been solved without difficulty. The highly sophisticated nature of REDUCE which gives it its wide applicability and power brings with it much scope for strange errors and obtuse messages. However the new release, REDUCE 3·2 is evidently more robust and has overcome many of these problems.

We approached symbolic computing in a cautious frame of mind. We first attempted to do small problems with solutions that could also be obtained manually with reasonable ease. We then proceeded to use REDUCE to check a very long and complicated analysis as described above. Only when we had achieved term for term agreement with this calculation did we feel happy to use the technique alone. Although there are certain features of our present system (REDUCE 3·0 implemented on an IBM 3033 operating CMS running under VM/SP release 2) that do not work, we have not discovered any way of obtaining erroneous results. Discussions with other users have also failed to bring any such problems to light. Our careful approach was therefore unnecessary which is no surprise since REDUCE has been in use in high energy physics for many years.

6. RESULTS FROM THE WIND TURBINE CALCULATIONS

It is hoped that readers of this paper will be interested in both the application of symbolic computing to engineering problems and in the problems themselves. The original reason for embarking on the analyses described here was to ascertain the stability characteristics of wind turbine designs. It is well known that helicopters, which have some similarity with wind turbines, are prone to severe mechanical and aeroelastic instabilities. Wind turbines seem less susceptible to these problems, but it is nevertheless prudent to check for possible unstable regions within the operational envelope.

It was pointed out earlier that the simple three degree-of-freedom system used as an example above had dynamic characteristics which had more in common with helicopters than wind turbines. This comment is confirmed by consideration of the results presented in Figure 3. These were computed by using standard Floquet techniques together with some more economical methods; these calculations have been described by Quarton and Garrad [14]. They clearly show a region of instability. Investigation of the instability reveals it to be a "ground resonance" type. Such an instability requires the blades to be very soft in the lead-lag direction. The present generation of wind turbines tend to have blades that are very stiff in this direction and hence this type of instability cannot occur.

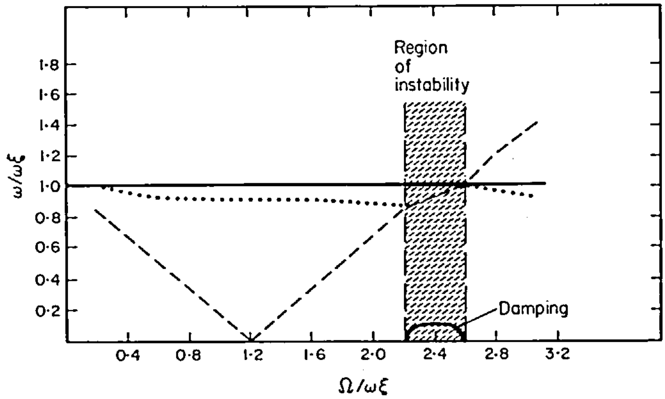

Figure 3. System frequencies of three degree of freedom model. ——, Rotor collective mode; - - -, rotor cyclic mode; · · · , tower lateral mode.

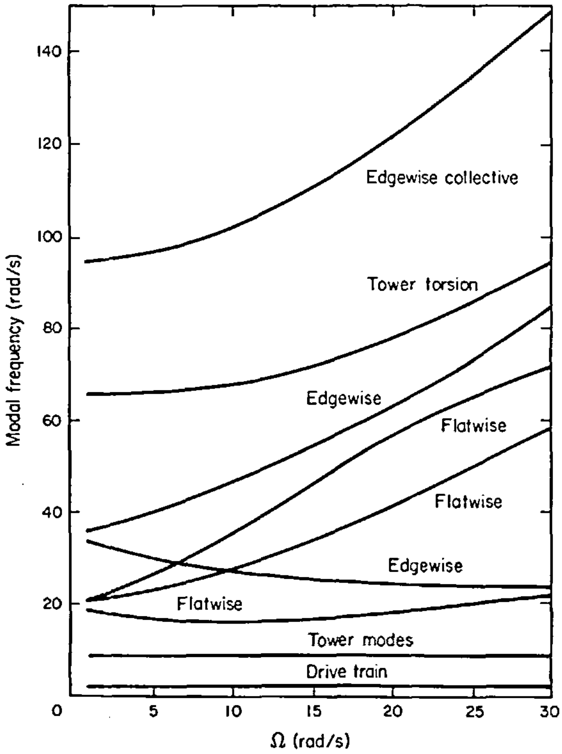

Figure 4. System WEG MS-2 frequencies.

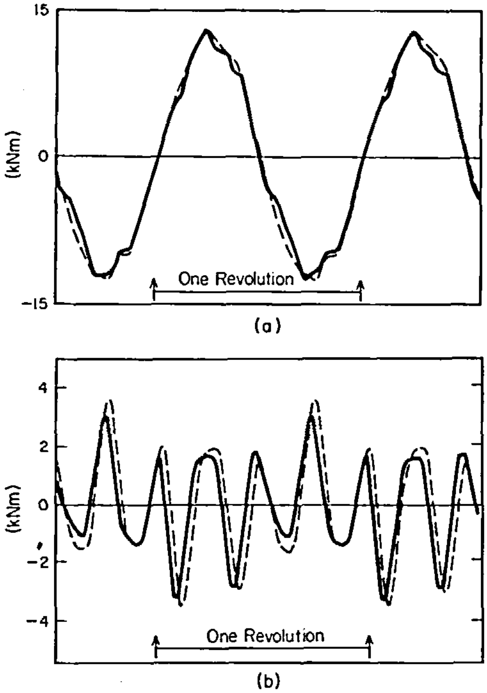

Figure 5. Predicted and measured WEG MS-1 blade bending moments at 16 m/s windspeed. ——, Predicted; - - -, measured. (a) Edgewise bending moment at 3·28 m; (b) flapwise bending moment at 3·28 m.

A more typical set of stability results is shown in Figure 4. These were obtained for the WEG MS-2 wind turbine (a 25 m diameter, 3-bladed, 200 kW machine). This model contains the fundamental flapping and lagging modes of each blade, a rigid body drive train freedom, and the first fore-aft, side-to-side and torsional modes of the tower. The eigenvalues of the system were obtained from a constant coefficient set of equations which resulted from performing a Coleman multi-blade transformation on the set of periodic coefficient equations as described above in section 3.4. The equations of motion of the system were derived entirely by REDUCE which also wrote a considerable part of the FORTRAN program used to calculate the numerical results.

In addition to using REDUCE for the stability analysis of wind turbine systems it has also been used for the derivation of mathematical models for the calculation of forced response. The procedure for deriving the equations of motion for a forced response model is essentially the same as for stability analysis except it then becomes important to retain all steady forcing and deflection terms as well as those which are proportional to the system generalized co-ordinates.

A forced response model for the WEG 20 m diameter, 2-bladed, 250 kW MS-1 wind turbine has been derived by using REDUCE and typical response predictions are presented in Figure 5. These predictions of blade bending moment have been compared with equivalent measured results and their satisfactory agreement provides validation of the mathematical model.

7. CONCLUDING REMARKS

The purpose of this paper was to describe one area of engineering analysis in which symbolic computing has played a useful role. It is hoped that the examples described here together with the specific comments on the use of the REDUCE system will enable other analysts to consider the possibility of using symbolic computing to help tackle complicated algebraic problems.

ACKNOWLEDGMENTS

The authors wish to thank the U.K. Department of Energy and the Directors of Taylor Woodrow Construction, both of whom have supported parts of this work, for their permission to publish this paper. We have received invaluable help over our problems with REDUCE from Dr Pat Pearce of Kingston Polytechnic and Professor John Fitch of the University of Bath. We are also very much in debt to Rob Warren of Taylor Woodrow Construction and Dr Larry Seward of the Rand Corporation, who have spent a lot of time and energy implementing the REDUCE system on Taylor Woodrow's IBM 3033.

REFERENCES

1. H. G. KAHRIMANIAN 1953 *MS Thesis, Temple University, Philadelphia, Pennsylvania,* Analytical differentiation by digital computer.
2. J. NOLAN 1953 *SM Thesis, MIT Cambridge, Massachusetts.* Analytical differentiation on a digital computer.
3. J. NAGABHUSHANAM, G. H. GAONKAR and T. S. R. REDDY 1981 *Proceedings of the Seventh European Rotorcraft and Powered Lift Aircraft Forum, Garmisch-Partenkirchen, Federal Republic of Germany, Paper No* 37. Automatic generation of equations for rotor body systems with dynamic inflow for *a priori* ordering schemes.
4. A. K. NOOR and C. M. ANDERSON 1979 *Computers and Structures* **10**, 95–118. Computerised symbolic manipulation in structural mechanics progress and potential.

5. R. PAVELLE, M. ROTHSTEIN and J. FITCH 1981 *Scientific American*, 101-113. Computer algebra.

6. J. JENSEN and F. NIORDSEN 1977 in *Structural Mechanics Software Series* (Editors N. Perrone and W. Pilkey) Charlottesville: University Press of Virginia. Symbolic and algebraic manipulation languages and their applications in mechanics.

7. F. KIESSLING 1982 *ICAS Proceedings*, 13*th Congress of the International Council of the Aeronautical Sciences/AIAA Aircraft Systems and Technology Conference, Seattle, Washington.* Computer-aided derivation of equations of motion for rotary-wing aeroelastic problems.

8. F. KIESSLING 1984 *DFVLR Forschunsbevicht* **84-10**, 187 pages. Modelling of the overall aeroelastic system of a wind turbine aided by symbolic programming. (In German.)

9. W. P. KOPPENS 1983 *Technische Hogeschool Delft, Luchtvaart en Ruimtevaarttechnik, Memorandum* M477. Utilization of the symbolic manipulation system REDUCE 2 for deriving equations of motion.

10. J. FITCH 1985 *Journal of Symbolic Computation* **1**, 211-227. Solving algebraic problems with REDUCE.

11. J. DUGUNDJI and J. H. WENDELL 1983 *American Institute of Aeronautics and Astronautics Journal* **21**, 890-897. Some analysis methods for rotating systems with periodic coefficients.

12. H. H. OTTENS and R. J. ZWAAN 1978 *NLR TR* 78115 *L*. Description of a method to calculate the aeroelastic stability of a two-bladed horizontal axis wind turbine.

13. K. R. V. KAZA, D. C. JANETZKE and T. L. SULLIVAN 1980 *Journal of Energy* **4**, 162-169. MOSTAS computer code evaluation for dynamic analysis of two-bladed wind turbines. (Also *NASA TM* 79101.)

14. D. C. QUARTON and A. D. GARRAD 1984 *Proceedings of the* 6*th British Wind Energy Association Conference*, pp. 197-209. Some comments on the stability analysis of horizontal axis wind turbines.