Automatic Generation of Equations of Motion from Graphic Input of Vibration Model*

(A System for Two-Dimensional Vibrations of Multibody Systems)

Keiichi MARUYAMA** and Takafumi FUJITA***

In order to make vibration analysis an easily input, automated task for designers, it is desirable to develop an interactive vibration analysis system, using personal computers or engineering work stations which have a user-friendly interface for easy input of the vibration model, and which can perform automatic generation of equations of motion, and automatic numerical analysis of them. As the first step towards such a system, an automatic generation system of equations of motion was developed for two-dimensional vibrations of multibody systems. The system is written in Smalltalk-80, which is a typical object-oriented language having the following advantages: (1) user-friendly interface for graphic input of a vibration model, enabling users to make a variety of configurations of a vibration model on the display by choosing the vibration model elements listed in the menu with a pointing device (mouse); (2) applicability to extensive models of two-dimensional vibrations of multibody systems which contain various constraints and nonlinear elements.

Key Words: Vibration, Automatic Generation of Equations of Motion, Vibration Model, Graphic Input, Object-oriented Language

1. Introduction

In recent years, the finite-element method using automatic mesh generation for the shape model in CAD, and experimental mode analysis by vibration tests on real machines have become widely used for automating and reducing of the amount of work involved in vibration analysis of machines or structures. For certain types of machines or structures, however, designers continue to use traditional methods of vibration analysis, in which they make a vibration model, derive its equations of motion, and investigate the characteristics of the vibrations by numerical or theoretical analysis. An automated or labor-saving vibration analysis is therefore desirable. An interactive vibration analysis system that meets this requirement by enabling users of the system to input a vibration model easily, and by deriving and

analyzing equations of motion automatically, would be very useful.

In this research, as the first step towards such an interactive vibration analysis system, a system was developed that can generate equations of motion from user's graphic input of a two-dimensional vibration model of multibody systems with a pointing device (mouse).

2. System Configuration

The main hardware configuration is a PC-9801VM2 (NEC), a 20MB hard disk, and a mouse. A 68000 CPU and a 2MB memory for the CPU are on the extension slot. The programming language for the system is Smalltalk-80 which is a typical object-oriented language.

Object-oriented languages have the uniqueness that they give autonomy to data, which usually have a subordinate nature, by including programs in the data itself. Such autonomous data patterns are called objects. In order to make an object perform a procedure (method), a message is sent to it. The same kind of objects are classified together in the same class. Messages and methods to be sent are described in the

*Received 21st September 1987. Paper No. 87-0065A

**Tokyo Research Laboratory, IBM Japan, Ltd., 5-19 Sanbancho, Chiyoda-ku, Tokyo, 102, Japan

***Institute of Industrial Science, University of Tokyo, 7-22-1 Roppongi, Minato-ku, Tokyo, 106, Japan

both ends of the connection. Rotation restraints equalize the rotations of rigid bodies on both ends of the connection. Boundaries move only in the directions listed in the menu and should be connected to matching elements. When a datum plane is declared, the motion of other elements is taken as observed from the datum plane. Otherwise, the motion of elements is taken as observed from an inertial coordinate system. The origin of the displacement of each element is the point of static equilibrium. The system assumes that the motion of elements is so slight that it is independent in each direction.

3.2 Method of input

The system uses two windows for the input of a vibration model and the display of its equations of motion. The first operation is to open the window for graphic input of the vibration model (a kind of graphic editor). This is done by selecting the item listed in the pop-up menu as "System Configuration" by pushing the mouse button in an empty part of the screen.

Figure 2 shows the method of inputting a vibration model. In this figure, the vibration elements are first placed in the window. Vibration elements can be selected in the window's pop-up menu (Fig. 3). The display of the selected element is moved in accordance

Fig. 2 Example of input of a vibration model

rigid body
particle
connection point
X spring
X damper
Y spring
Y damper
R spring
R damper
X restraint
Y restraint
R restraint
X boundary
Y boundary
R boundary
datum plane
X force
Y force
moment
X nonlinear
Y nonlinear
connect
equation

Fig. 3 Menu of vibration elements

with the motion of the mouse. It is fixed on the window by clicking the mouse button. The index of the element is then automatically allocated and displayed.

The connection of elements is done by selecting the "connect" item and clicking the mouse button on the elements. A vibration model is perfected by repeating these operations.

There are, however, a few exceptions. A rigid body is fixed by designating a rectangular area in the same way as when opening a window. A rigid body cannot be directly connected to other elements, such as springs and dampers. Connection points must be arranged on a rigid body. Connection points and other elements can be connected. A connection point is displayed on a monitor by selecting the "connection point" item listed in the pop-up menu and clicking the mouse button in a rectangle representing a rigid body. An external force must be fixed on a point of action that is a particle or a connection point. A moment to a rigid body can be fixed at any place on the rigid body, because the place in the rigid body does not affect the equations of motion. When selecting nonlinear elements, the "x nonlinear" or "y nonlinear" item should be selected, depending on the direction. The graphic menu (Fig. 4) will then appear and the necessary nonlinear element can be selected. The system adopts four elements in order to show how to deal with nonlinear elements. However, for practical use, we believe that it is necessary to prepare several nonlinear elements and a method of input enabling users to define new nonlinear elements easily. To declare a datum plane, it is necessary to select the "datum plane" item and to click the mouse button on the boundary.

In order to generate the equations of motion of

Fig. 4 Menu of nonlinear elements

the input model, the window for displaying equations of motion must be opened. This is done by selecting the "equation" item listed in the menu (Fig. 3). Then, by selecting it again, equations of motion are generated automatically.

The sequence of these operations makes the system generate equations of motion, although the system has no modes of operation. Hence it is not necessary that the sequence be the same as in the example in Fig. 2. A vibration model can be freely modified, and the system can generate new equations of motion each time.

4. The Basic Algorithm for Generation of Equations of Motion

4.1 For particles

The equations of motion of vibration models are uniquely determined by simply specifying the connecting relation among the vibration elements. The system prepares an instance of Multibody Vibration as an object that recognizes all input elements. Each element-object has instance variables for necessary data, such as collections of element-objects, connected to it. Figure 5 shows an outline of the message-flow for generation of the equations of motion of particles. Automatic generation of equations of motion starts by sending messages to the particles requesting their equations (①). The particle that is sent the message starts generating a string representing an equation of motion. An inertia term is generated by connecting a string representing a mass ('M1') to a string representing an absolute acceleration. A particle has strings representing displacement and velocity. Acceleration can be derived from the velocity. For an inertial coordinate system, by converting 'dy' (that is, a string for velocity) to 'ddy', an absolute acceleration can be derived ($dy \rightarrow ddy$). When a datum plane is declared, as in Fig. 5, a relative acceleration is derived by that operation and an absolute acceleration can be derived by adding the absolute acceleration of the datum plane to it ($ddy1 + ddyb1$). Damping force terms and elastic force terms (③, ⑤) can be derived

Fig. 5 Message-flow for particles

by sending messages (②, ④) to springs and dampers connected to a particle. Dampers and springs generate and return the forces (③, ⑤) by deriving relative velocity terms and relative displacement terms from the velocity and displacement between both ends of the connection. The left side of an equation is completed by the above-mentioned operations. The right side of an equation is a collection of external force terms if external forces on a particle are registered, or '0' if no forces are registered. This is the method for the generation of equations of motion for a system of particles.

4.2 For Rigid bodies

The displacement information in a rigid body exists in its connection points. Therefore, a rigid body's instance variables are only collections of connection points and moments. Connection points link collections of vibration elements to a rigid body.

Figure 6(a) shows an outline of the message-flow for generation of the equations of motion of a rigid body for translation (x direction). An instance of Multibody Vibration sends 'J1' a message (①) requesting an equation of motion for the x direction. 'J1'

(a) translation

(b) rotation

Fig. 6 Message-flow for rigid bodies

lar acceleration) are placed together because each (angular acceleration) has the same value in all cases. The moment of inertia in this case is displayed as a string of summations of the moments of inertia of all the rotationally restrained bodies.

5.2 For Translational Restraints

Fig. 8 Message-flow of update of a rotational angle for rotational restraints

Fig. 9 Message-flow of derivation of inner forces for translational restraints

As preprocessing for the automatic generation of equations of motion, the update of displacement and velocity is done in a similar way to that described in section 5.1. The displacement and velocity of any point in a rigid body are derived by adding the displacement and velocity caused by rotation to the displacement and velocity of the center of gravity of the rigid body, respectively. The preprocessing is done in the following manner. First, the displacement and velocity of the source point (a particle or a connection point) are derived. If the receiver is a particle, they are registered as its displacement and velocity. If the receiver is a connection point, they are revised as the displacement and velocity of the center of gravity of the rigid body to which the connection point belongs, then registered to the rigid body.

The inner forces caused by translational restraints affect equations of rotation. They must therefore be acquired and registered as preprocessing for the generation of the equations. Figure 9 shows an outline of the derivation of inner forces. Message (①) from an instance of Multibody Vibration causes messages (②, ③) to be sent to the end of a tree structure of the restrained bodies. When the end ('J2') is found (④), the inner force (⑤) is derived and registered to 'P4' and then the reverse force (⑥) is sent and registered to 'P3' which is the counterpart of 'P4'. The following process is similar and all the inner forces are subsequently derived and registered.

The system derives inner moments in the above-mentioned manner.

Figure 10 shows a hard copy of execution for a

Fig. 10 Example of a linear system with restraints

linear vibration model with restraints. In this example, the system displays inner forces and inner moments, as well as equations. The system also displays at which points, in which directions the inner forces operate and on which rigid bodies the inner moments operate.

6. The Algorithm for Nonlinear Elements

6.1 For nonlinear elements without memories

The nonlinear damper $\{A|\dot{x}|B\text{sgn}(\dot{x})\}$ is established by modifying the expression for the force of the linear damper, as shown in Fig. 11. The system expresses the force of a nonlinear spring with backlash as a function with an argument, the relative displacement

Fig. 11 Example of a nonlinear system with nonlinear elements without memories and elasto-plastic hysteresis elements

Fig. 12 Example of a nonlinear system with friction elements

ment. The system then defines the function separately (see Fig. 11).

6.2 For The elasto-plastic hysteresis element

The system adopts the formalization of an elasto-plastic hysteresis element by differential equations and uses the following method described in reference (3).

$$\dot{H}(t) = K(v)[1 - \text{sgn}(v)\{H(t)/D - T\}^N]$$

$$T = E\{v/U - H(t)\}/D$$

$$K = D/U$$

' $H(t)$ ' is the elastic force of the elasto-plastic hysteresis element. ' v ' is the relative velocity of its ends. ' K ' is the initial rigidity. ' D ' is the yield force. ' U ' is the yield displacement. ' E ' is a parameter for the gradient after yield. ' N ' is a parameter for a hysteresis loop figure when the state of the element is changing from elasticity to plasticity.

The system expresses merely the corresponding force in equations and displays differential equations for derivations of it (see Fig. 11).

6.3 For friction elements

Both kinetic friction and static friction are considered in the system. Friction elements can be in a "sliding" or a "nonskid" state. When the number of friction elements is n , the number of phases of the whole system is 2^n . The system defines the relations between the state of each element and the phase number by bit manipulation. If the n -th bit of the phase number is zero, it indicates that the n -th friction element does not slide.

If a friction element does not slide, it restrains the motion of translation, keeping a relative displacement between the ends of the connection. In the generation of equations, the system ignores forces generated by the nonskid friction elements and other elements being parallel to them. The nonskid friction elements are processed as elements of restraints by other element-objects in the system. However, the displacements are not updated.

The switching condition between two phases is a combination of the switching conditions when each friction element changes to another state. Therefore, the system sends messages to all friction elements and obtains their switching conditions when generating equations for each phase.

When a friction element changes from the sliding state to the nonskid state, the system derives the force acting on the element. The switching condition is the logical combination of the condition that the force is less than the static friction force, and the condition that the relative velocity between the ends is zero.

When a friction element changes from the non-skid state to the sliding state, the force acting on the element is obtained in the process of deriving inner

forces, because the force is balanced with the sum of an inner force and an elastic force due to the relative displacement. The switching condition is that the force is more than the static friction force. After all

Equations of Motion

Phase1

Inner force

$$\begin{aligned} &M1ddybl \quad (M1 \ Y) \\ &-M1ddybl \quad (M2 \ Y) \\ &M2ddybl + M1ddybl \quad (M2 \ Y) \end{aligned}$$

Phase2

$$\begin{aligned} &M1ddy1 \\ &+Kyl(y1-y2)+Syl \ \text{sgn}(dy1-dybl) \\ &=0 \end{aligned}$$

Inner force

$$M2ddybl + Kyl(y2-y1) + Syl \ \text{sgn}(dybl-dy1) \quad (M2 \ Y)$$

Phase3

$$\begin{aligned} &M1ddy1 \\ &+M2ddy1 \\ &+Sy2 \ \text{sgn}(dy1-dybl) \\ &=0 \end{aligned}$$

Inner force

$$\begin{aligned} &-M2ddy1 - Sy2 \ \text{sgn}(dy1-dybl) \quad (M1 \ Y) \\ &M2ddy1 + Sy2 \ \text{sgn}(dy1-dybl) \quad (M2 \ Y) \end{aligned}$$

Phase4

$$\begin{aligned} &M1ddy1 \\ &+Kyl(y1-y2) + Syl \ \text{sgn}(dy1-dy2) \\ &=0 \end{aligned}$$

M2ddy2

$$\begin{aligned} &+Kyl(y2-y1) + Syl \ \text{sgn}(dy2-dy1) + Sy2 \ \text{sgn}(dy2-dybl) \\ &=0 \end{aligned}$$

Phase1→Phase2

$$M1ddybl + Kyl(y1-y2) \mid > Syls$$

Phase1→Phase3

$$M2ddybl + M1ddybl \mid > Sy2s$$

Phase1→Phase4

$$M1ddybl + Kyl(y1-y2) \mid > Syls$$

$$\& \mid M2ddybl + M1ddybl \mid > Sy2s$$

Phase2→Phase1

$$M1ddy1 + Kyl(y1-y2) \mid < Syls \ \& \ dy1-dybl=0$$

Phase2→Phase3

$$M1ddy1 + Kyl(y1-y2) \mid < Syls \ \& \ dy1-dybl=0$$

$$\& \mid M2ddybl + Kyl(y2-y1) + Syl \ \text{sgn}(dybl-dy1) \mid > Sy2s$$

Phase2→Phase4

$$M2ddybl + Kyl(y2-y1) + Syl \ \text{sgn}(dybl-dy1) \mid > Sy2s$$

Phase3→Phase1

$$M1ddy1 + M2ddy1 \mid < Sy2s \ \& \ dy1-dybl=0$$

Phase3→Phase2

$$M2ddy1 + Sy2 \ \text{sgn}(dy1-dybl) + Kyl(y2-y1) \mid > Syls$$

$$\& \mid M1ddy1 + M2ddy1 \mid < Sy2s \ \& \ dy1-dybl=0$$

Phase3→Phase4

$$M2ddy1 + Sy2 \ \text{sgn}(dy1-dybl) + Kyl(y2-y1) \mid > Syls$$

Phase4→Phase1

$$M1ddy1 + Kyl(y1-y2) \mid < Syls \ \& \ dy1-dy2=0$$

$$\& \mid M2ddy2 + Kyl(y2-y1) + Syl \ \text{sgn}(dy2-dy1) \mid < Sy2s \ \& \ dy2-dybl=0$$

Phase4→Phase2

$$M2ddy2 + Kyl(y2-y1) + Syl \ \text{sgn}(dy2-dy1) \mid < Sy2s \ \& \ dy2-dybl=0$$

Phase4→Phase3

$$M1ddy1 + Kyl(y1-y2) \mid < Syls \ \& \ dy1-dy2=0$$

Fig. 13 Print-out of equations for the system in Fig. 12

of the switching conditions have been derived in each phase, the system processes a loop (in which the loop counter (k) is from 1 to $2^n - 1$) in order to derive all phases to be transitioned from a phase and conditions for the transitions. If each bit of the number (k) is 1, the system interprets it as the corresponding friction element being in an altered state (so " k " cannot be zero) and combines switching conditions according to necessity. A destination phase in transition is derived by XOR (exclusive OR) by each bit between the phase number of that stage and ' k '. In logical operation, phase numbers are from 0 to $2n - 1$. In this system, however, the phase number is incremented and displayed because Phase 0 is not desirable.

Figure 12 shows a hard copy of an example of execution. The object in the left part of the window for displaying equations is called a scroll bar, and it indicates the scope of the display. Users of the system can browse equations by scrolling with this scroll bar. The system can also print all the equations. Figure 13 shows a print-out of the equations of the model in Fig. 12. The system displays the equations one after another. In the case of Fig. 12, it takes about 2 seconds for the first equation to be displayed and about 1 min until the last switching condition is displayed.

7. Conclusion

In this research, as the first step towards an

interactive vibration analysis system, a system was developed with the following merits.

1. a good user interface for graphic input of a vibration model, enabling users to form and vary the configuration of a vibration model on the display easily by choosing the vibration model elements listed in the menu with a pointing device (mouse);
2. applicability to extensive models of two dimensional vibrations of multibody systems which contain various restraints and nonlinear elements.

However, the extensions to three-dimensional vibration models and other input methods for nonlinear elements remain as future research topics.

References

- (1) Goldberg, A. and Robson, D., *Smalltalk-80 The Language and its Implementation*, (1983), p. 1, Addison-Wesley.
- (2) Goldberg, A., *Smalltalk-80 The Interactive Programming Environment*, (1984), p. 1, Addison-Wesley.
- (3) Bhatti, M. A. and Pister, K. S., A Dual Criteria Approach for Optimal Design of Earthquake-Resistant Structural Systems, *Int. J. Earthqu. Eng. Struct. Dyn.*, Vol. 9, No. 6 (1981), p. 557.
- (4) Kobayashi, F., Hara S., Warita K., and Harada J., A System for Mechanical Vibration Analysis Based on the Object-Oriented Programming, *proc. of the 25th SICE Annual Conf.*, 2 (1986), p. 907.