Vietnam Journal of Mechanics, NCST of Vietnam Vol. 21, 1999, No 1 (36 - 44)

AN ALGORITHM FOR DERIVING EQUATIONS OF MOTION OF CONSTRAINED MECHANICAL SYSTEM

DINH VAN PHONG
Hanoi University of Technology, Vietnam

ABSTRACT. The article deals with the form of equations of motion of mechanical system with constraints. For holonomic systems the number of differential equation is equal to the degrees of freedom, without regard to the number of chosen coordinates. The possibilities of computer processing (symbolical and numerical) are shown. Two simple examples demonstrate the described technique.

1. Introduction

There are a lot of techniques for building equations of motion of mechanical systems. The conventional approaches could be divided into two groups, see e.g.[7], [8]. In the first one, a minimal set of Lagrangian variables, equal to a degree of freedom, is chosen, in order to define the system configuration. The number of differential equations is minimal and equal to a degree of freedom. The drawback of this technique is complexity of equations of motion and even the computer processing (e.g. by recursive algorithm) is time consuming. The second group of methods uses a larger number of coordinates in combination with constraints. The form of system of equations is simple, permitting computer generations. However, a final mixed system of differential-algebraic equations is large, including not only Lagrangian coordinates, but also so-called Lagrange multipliers. The total equations consist of differential equations which number is equal to number of chosen coordinates, and equations of constraints. In the present paper we will show that it is possible to derive the equations of motion with only a minimum of differential equations. Moreover there exists the possibility of calculation of reaction forces.

2. The form of equations of motion

Let us consider the dynamical system with m degrees of freedom. For this system we choose n Lagrangian coordinates q_i, $i = 1, 2, ..., n$ $(n \geq m)$, which are coupled by s constraint equations. In the present paper we consider only *ideal*

holonomic constraint conditions, which have the form:

$$g_i(q_1, q_2, \ldots, q_n, t) = 0, \quad i = 1, 2, ..., s \tag{2.1a}$$

or in the matrix form:

$$\mathbf{g}(\mathbf{q}, t) = 0, \tag{2.1b}$$

where $\mathbf{q} = \begin{bmatrix} q_1 & q_2 & \ldots & q_n \end{bmatrix}^T$ and $\mathbf{g} = \begin{bmatrix} g_1 & g_2 & \ldots & g_s \end{bmatrix}^T$. Clearly

$$m + s = n. \tag{2.2}$$

With redundan coordinates the equations of motions, as stated above, are derived very easily by using various techniques, e.g. by Lagrange equations of 2nd kind or by Newton-Euler equations. Here, they are not described explicitly and suppose that they have the following form for the system without constraints (2.1):

$$\mathbf{M}(\dot{\mathbf{q}}, \mathbf{q}, t)\ddot{\mathbf{q}} + \mathbf{h}(\dot{\mathbf{q}}, \mathbf{q}, t) = 0. \tag{2.3}$$

For particular cases of dynamical systems this equation could appear in various forms, accordingly the used methods. But in this article no detail discussion about them is devoted because the algorithm, derived below, does not depend on concrete form of equation (2.3). One should have only on remind that equations of motion are generated with redundant coordinates more easily than without them.

Now, due to constraint conditions (2.1) the equation (2.3) is not satisfied. Consequently, new quantities appear in the equation, see [4]:

$$\mathbf{M}(\dot{\mathbf{q}}, \mathbf{q}, t)\ddot{\mathbf{q}} + \mathbf{h}(\dot{\mathbf{q}}, \mathbf{q}, t) = \mathbf{r}, \tag{2.4}$$

where $\mathbf{r}$ is the vector of generalized reactions $\mathbf{r} = \begin{bmatrix} r_1 & r_2 & \ldots & r_n \end{bmatrix}^T$. In the mixed system of $(n + s)$ differential-algebraic equations (2.4) and (2.1) we have $2n$ unknowns: $\mathbf{r}$ and $\mathbf{q}$. In order to close the problem we should look for another $(n - s)$ equations.

In the case of mechanical system with ideal constraint condition we will have these equations in this form, see [2]:

$$\mathbf{D}^T \cdot \mathbf{r} = 0, \tag{2.5}$$

where $\mathbf{D}$ is a matrix of dimension $n \times m$. This matrix is derived from the criterion of ideality of the constraints:

$$\mathbf{G} \cdot \mathbf{D} = 0, \tag{2.6}$$

where G is Jacobian of constraints (2.1)

$$\mathbf{G} = \left[\frac{\partial g_i}{\partial q_j}\right]_{\substack{i=1,2,\ldots,s \\ j=1,2,\ldots,n}}$$

Obviously the dimension of G is $s \times n$.

The technique for finding D from (2.6) will be discussed far below. At this point, suppose that we have D defined. So, the system of (2.1), (2.4), (2.5) has really $2n$ equations for $2n$ unknowns r and q.

We can see that r is necessary for calculating reaction forces, but for integrating process these quantities make the order of the system of differential equations larger. Moreover r appears in the system without derivation. It seems to be more reasonable to provide integrating process without these quantities and calculate them after integrating process.

Removing r from (2.4) and replacing it into (2.5) yield:

$$\mathbf{D}^T \cdot \left(\mathbf{M}(\dot{\mathbf{q}}, \mathbf{q}, t)\ddot{\mathbf{q}} + \mathbf{h}(\dot{\mathbf{q}}, \mathbf{q}, t)\right) = 0. \tag{2.7}$$

This is a final differential equation of motion that we have look for. This new form of equations of motion gives only m differential equations for the system of m degrees of freedom. Obviously, system (2.7) and (2.1) have completely n=m+s equations for n unknowns q. In order to see, how interesting the form (2.7) is, we write it in the scalar form:

$$\sum_{i=1}^{n} d_{ij}\Big(\sum_{k=1}^{n} m_{ik}\ddot{q}_k + h_i\Big) = 0 \quad \text{for } j = 1, 2, \ldots, m. \tag{2.8}$$

It should be emphasised again that equation (2.7) does not depend on the way how the matrix equations (2.3) is generated. For example, if the equation (2.3) is derived from Lagrange equation of 2nd kind, we can write:

$$\sum_{i=1}^{n} d_{ij}\left[\frac{d}{dt}\Big(\frac{\partial T}{\partial \dot{q}_i}\Big) - \frac{\partial T}{\partial q_i} - Q_i\right] = 0 \quad \text{for } j = 1, 2, \ldots, m. \tag{2.9}$$

Another example is the principle of compatibility, showed in [3], as the method for generating equations of motion (2.3). In this particular case when the matrix M is function only of q, one gets similarly:

$$\mathbf{D}^T \cdot \left(\mathbf{A}(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \mathbf{h}_1(t, \mathbf{q}, \dot{\mathbf{q}}) + \mathbf{h}_2(\mathbf{q}, \dot{\mathbf{q}})\right) = 0, \tag{2.10}$$

where $\mathbf{A}(\mathbf{q})$ is inertia matrix of mechanical system, $\mathbf{h}_1$ and $\mathbf{h}_2$ are vectors of dimension n.

Using system (2.7) and (2.1) we gain something against the conventional form by using Lagrange multipliers. At first, the total number of equations is not $n+s$ but only n, which is the number of chosen coordinates (imagine, it is the degree of freedom for the system without constraints). And the number of differential equations is only $m = n - s$, which is a minimum since it is exactly the degree of freedom of the system. The second advantage is that we have in the equations (2.7) and (2.1) only Lagrangian coordinates and nothing more. It makes finding initial conditions and integrating more easily. Of course, after integrating the evaluation of $\mathbf{r}$ is very easy from (2.4). And with them also physical reaction forces could be calculated.

For numerical solving the system (2.7) and (2.1), in special cases, we can use various techniques to provide integrating. But, in general, the most reasonable way is using some implicit formulas as implicit Runge-Kutta methods, or using Gear algorithms, see e.g. [6], [8], [9]. These algorithms allow us to solve more general and complex problems when the constraints are e.g. nonholonomic.

Now, returning to the equations (2.6), we will show how coefficients d_{ij} could be evaluated. Some elegant techniques, basing on intuition of the solver, could be provided. But we will concentrate on the computer processing.

The first method, described in [2], is numerical and bases on the solution of undetermined system of algebraic equations. This algorithm was tested in many applications and some of results, reached by using the algorithm, are shown, e.g., in [1], [4]. Since this technique is bases on purely numerical treatment, all matrices in equation of motion (2.7) are generated separately for each time node, with repeating the same numerical algorithm.

Here, in this article, we will suggest another algorithm that is suitable even for computer symbolical processing. The advantage of this technique against the first numerical method is that matrix $\mathbf{D}$ could be derived symbolically only once at the beginning of integration process. The equation of motion (2.7) has exact symbolical form and for each time node one should only provide valuation of particular forms.

The process of finding $\mathbf{D} = \left[\mathbf{d}_1\ \mathbf{d}_2\ \ldots\ \mathbf{d}_i\ \ldots\ \mathbf{d}_m\right]$, where $\mathbf{d}_i$ is a vector of dimension $n \times 1$, from $\mathbf{G} = \left[\mathbf{g}_i^T\right]$, $i = 1, 2, \ldots, s$, where $\mathbf{g}_i$ is a vector of dimension $n \times 1$, consists of two steps.

In the first step the Gram-Schmidt orthogonalization with is realised. This presents the linear combination of original constraints, in order to get orthonormal

vectors $\mathbf{b}_i$, $i = 1, \ldots, s$, more suitable for next manipulation. It could be also used for checking the redundancy of constraints $\mathbf{g}_i$.

In the second step we will find vectors $\mathbf{d}_i$, $i = 1, ..., m$, which together with $\mathbf{b}_i$ create an orthonormal system. They are evaluated consequently from $\mathbf{d}_1, \mathbf{d}_2, \ldots$ to $\mathbf{d}_m$.

The algorithm is as follows:

Step 1: For $i = 1$ to s do

$$\mathbf{b}_i = \mathbf{g}_i - \sum_{j=1}^{i-1} (\mathbf{g}_i^T \mathbf{b}_j)\mathbf{b}_j$$

$$\mathbf{b}_i = \frac{\mathbf{b}_i}{\|\mathbf{b}_i\|}$$

end of i-loop.

Step 2: For $i = 1$ to m do

$$\mathbf{d}_i = \mathbf{x}_i - \sum_{j=1}^{s} (\mathbf{x}_i^T \mathbf{b}_j)\mathbf{b}_j - \sum_{j=1}^{i-1} (\mathbf{x}_i^T \mathbf{d}_j)\mathbf{d}_j$$

$$\mathbf{d}_i = \frac{\mathbf{d}_i}{\|\mathbf{d}_i\|}$$

end of i-loop. •

In this process $\mathbf{x}_i$ is an arbitrary vector, different from any $\mathbf{b}_i$ or $\mathbf{d}_i$, already defined before. Symbol $\|\mathbf{a}\|$ denotes the Euclidean norm of vector $\mathbf{a}$.

Note that we get vectors $\mathbf{d}_i$ that are orthonormal, but this condition is required in our algorithm only for easier manipulation and not from (2.6). By multiplying various scales of $\mathbf{d}_i$, matrix $\mathbf{D}$ could take the most convenient form.

The last remark is about equation (2.5). For other dynamical cases from technical life, such as for controlled systems or the system with non-ideal constraints, this equation is replaced by other ones. And this is another advantage of described technique against conventional approaching.

Now consider two simple examples to illustrate described form of equations of motion. The processing seems to be time consuming and not so easily for a man, but such software with symbolical manipulation as MAPLE or MATHEMATICA etc., will be useful tool for these cases.

Example 1. Consider the planar case of a rolling disk without slipping. The system has only one degree of freedom. But we choose three Lagrangian coordinates s, x and φ, as shown in Fig. 1. Clearly, $n = 3$, $s = 2$, and $m = 1$.

The constraint conditions are:

$$u - r = 0, \tag{3.1}$$

$$s + r\varphi = 0, \tag{3.2}$$

where r is a radius of the disk.

Obviously, we get the matrix G:

$$\mathbf{G} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & r \end{bmatrix}. \tag{3.3}$$

With 3 coordinates s, u and φ one gets easily the following equations:

$$m_d\ddot{s} - f(t) = r_s, \tag{3.4}$$

$$m_d\ddot{u} + m_d g = r_u, \tag{3.5}$$

$$J_d\ddot{\varphi} = r_\varphi, \tag{3.6}$$

where r_s, r_u, r_φ are the elements of vector of reaction forces $\mathbf{r}$:

$$\mathbf{r} = \begin{bmatrix} r_s & r_u & r_\varphi \end{bmatrix}^T.$$

Matrix D could be found by algorithm, described above in section 2.

In the first step one gets:

$$\mathbf{b}_1 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T,$$

$$\mathbf{b}_2 = \begin{bmatrix} \frac{1}{\sqrt{1+r^2}} & 0 & \frac{r}{\sqrt{1+r^2}} \end{bmatrix}^T.$$

The second step, with e.g. $\mathbf{x}_i = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$, yields the vector $\mathbf{d}_i$ (not normalised yet):

$$\mathbf{d}_i = \begin{bmatrix} \frac{r^2 - r}{1+r^2} & 0 & \frac{1-r}{1+r^2} \end{bmatrix}^T.$$

So the choice $\mathbf{D}^T = \begin{bmatrix} r & 0 & -1 \end{bmatrix}$ gives the equation (2.5) in the form:

$$r_s \cdot r - r_\varphi = 0. \tag{3.7}$$

By replacing r_s and r_φ from (3.4)-(3.6) into (3.7), finally we have the differential equation of motion (2.7) in the form:

$$m_d r\ddot{s} - rf(t) - J_d\ddot{\varphi} = 0. \tag{3.8}$$

And the system of equations (3.1), (3.2), (3.8) could be solved, in order to get **all quantities** $s, u, \varphi, \dot{s}, \dot{u}, \dot{\varphi}, \ddot{s}, \ddot{u}, \ddot{\varphi}$. After that, if desired, r_s, r_u and r_φ could be **evaluated** easily from (3.4), (3.5) and (3.6).

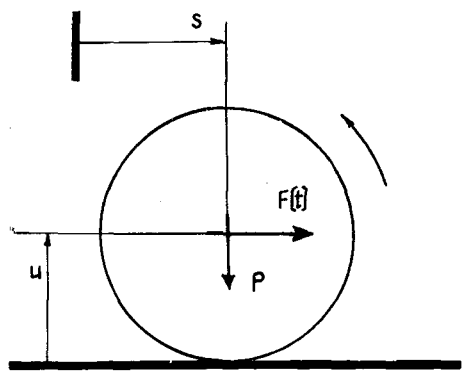

Fig. 1

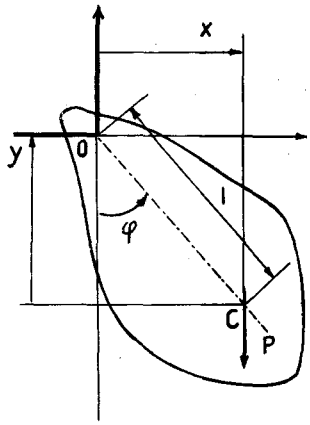

Fig. 2

Example 2. Consider a planar single pendulum with 3 coordinates x, y and φ as shown in Fig.2. So $n = 3$, $s = 2$, $m = 1$. Constraint conditions are:

$$x - \ell \sin\varphi = 0, \tag{3.9}$$

$$y + \ell \cos\varphi = 0, \tag{3.10}$$

and Jacobian **G** is:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & -\ell\cos\varphi \\ 0 & 1 & -\ell\sin\varphi \end{bmatrix}. \tag{3.11}$$

Similarly, as in example 1, one gets easily equations:

$$m_p\ddot{\mathbf{x}} = r_x, \tag{3.12}$$

$$m_p\ddot{\mathbf{y}} + m_p g = r_y, \tag{3.13}$$

$$J_p\ddot{\varphi} = r_\varphi, \tag{3.14}$$

And from the criterion of ideality, two steps of above described algorithm follow:

Step 1

$$\mathbf{B} = \begin{bmatrix} \dfrac{\ell}{\sqrt{1+\ell^2\cos^2\varphi}} & 0 & \dfrac{-\ell\cos\varphi}{\sqrt{1+\ell^2\cos^2\varphi}} \\ \dfrac{-\ell^2\sin\varphi\cos\varphi}{\sqrt{1+\ell^2}\sqrt{1+\ell^2\cos^2\varphi}} & \dfrac{\sqrt{1+\ell^2\cos^2\varphi}}{\sqrt{1+\ell^2}} & \dfrac{-\ell\sin\varphi}{\sqrt{1+\ell^2}\sqrt{1+\ell^2\cos^2\varphi}} \end{bmatrix}.$$

Step 2. A choice of a vector $\mathbf{x}_i = [0\ 1\ 0]^T$ yields:

$$\mathbf{D} = \frac{1}{1+\ell^2}\begin{bmatrix} \ell\sin\varphi\cos\varphi \\ \ell\sin^2\varphi \\ \sin\varphi \end{bmatrix}.$$

The simplest form of $\mathbf{D}$ is $[\ell\cos\varphi\ \ell\sin\varphi\ 1]^T$. So the equation (2.5) has the following form:

$$r_x\ell\cos\varphi + r_y\ell\sin\varphi + r_\varphi = 0. \tag{3.15}$$

And from (3.12), (3.13) and (3.14) we get finally the differential equation of motion (2.7) for our case:

$$\ell m_p\cos\varphi\ddot{x} + \ell m_p(\ddot{y}+g)\sin\varphi + J_p\ddot{\varphi} = 0. \tag{3.16}$$

Again, one can solve the mixed system of equations (3.16), (3.9) and (3.10) and get all required quantities of the considered mechanical system.

Conclusion

We have shown the form of differential equations of motion (2.7) for the mechanical system with n coordinates. The total number of equations (differential and algebraic) is logically equal to the number of coordinates n. For uncoupled system all n differential equations are presented to describe the system motion. Coupling the coordinates by constraints reduces the number of differential equations. Instead, we dispose the constraint equations. So we have as many differential equations of motion as degrees of freedom, i.e. the minimal number.

The key point for writing equation (2.7) is deriving the matrix $\mathbf{D}$. The algorithm, described in section 2, is suitable for computer processing with symbolical or numerical manipulation. For simple cases it is possible to provided it directly by hand too. But our main aim with this algorithm is to show a possibility to create one computer software for automatic generation the equation of motion of mechanical system with constraints.

The most advantages of this form of equations against conventional approaching are reduced number of equations and removing the quantities like Lagrange multipliers from integrating process. On the other hand, the possibilities of evaluating the quantities, useful for calculating reaction forces, are respected.

The paper is completed with financial support of the Council for Natural Science of Vietnam.

REFERENCES

1. Dinh Van Phong. Direct using constraint equations of mechanical systems. Proceedings of the international conference on applied dynamics, Hanoi, 1995, pp 79-85.
2. Dinh Van Phong. Principle of compatibility and criteria of ideality in study of constrained mechanical systems. Strojnicky casopis, 47, 1996, No.1, pp 2-11,Bratislava. (Journal of czech and slovak mechanical engineering)
3. Do Sanh. A new form of equations of motion of nonholonomic mechanical systems. Journal of Mechanics, NCNST of Vietnam, T. XIX, 1997, No 3 (59-64).
4. Do Sanh, Dinh Van Phong. Principle of compatibility and design of grinding machine. In: Education, practice and promotion of computational methods in engineering using small computers. Ed. Oliveira and Bento, Techno-Press, Korea,1995, Vol. 2, pp 1233-1237.
5. Do Sanh, Dinh Van Phong. The principle of compatibility and computational mechanics. Proceedings of the NCST of Vietnam, Vol. 7, No. 1, 1995.
6. Gear C. W. Differential-algebraic equations. In: Computer aided analysis and optimization of mechanical system dynamics. Edited by E. J. Haug. Springer-Verlag, Berlin, 1984.
7. Haug J. E. Computer aided kinematics and dynamics of mechanical systems. Volume 1: Basic methods. Allyn and Bacon, Massachusetts, 1989.
8. Nikravesh P. E. Computer aided analysis of mechanical systems. Prentice-Hall, Englewood Cliffs, 1988.
9. Roger K. A., James J. C. Runge-Kutta methods and differential-algebraic system. SIAM, Numer. Anal., Vol. 27, No. 3, June 1990, pp 736-752.

Received November 15, 1998

THUẬT GIẢI CHO MỘT DẠNG PHƯƠNG TRÌNH CHUYỂN ĐỘNG CỦA HỆ CƠ HỌC CHỊU LIÊN KẾT

Bài báo đề cập đến một dạng của phương trình chuyển động của cơ hệ chịu liên kết. Đối với cơ hệ hôlônôm dạng phương trình này chỉ có số phương trình vi phân bằng đúng số bậc tự do của cơ hệ, bất kể số tọa độ suy rộng dư được chọn là bao nhiêu. Mục đích của bài báo là chỉ ra khả năng thiết lập các phương trình này tự động bằng máy tính (cả dạng số và biểu thức). Hai ví dụ đơn giản được dùng để minh họa cho thuật giải được đề xuất.

Department of Applied Mechanics, Hanoi University of Technology
1 Dai Co Viet Str., Hanoi, Vietnam
Tel./Fax: (00).(84).(4).9780799 P.O Box: 435 Bo ho, Hanoi, Vietnam