2. A. Ye. Kobrinskiy and Yu. A. Stepanenko, Some problems in manipulator theory, In coll.: Mekhanika Mashin (Computer Mechanics), No. 7, 8, Nauka, 1967.
3. M. B. Ignat'yev, F. M. Kulakov and A. M. Pokrovskiy, Algoritm Upravleniya Robotami-Manipulyatorami (Robot-Manipulator Control Algorithm), Mashinostroyeniye, 1972.
4. Integral'nyye Roboty (Integrated Robots), Coll. of papers edited by G. Ye. Pozdnyak, Mir, 1973.
5. G. Korn and T. Korn, Spravochnik po Matematike (Mathematics Handbook), Nauka, 1973, (Translated from English).
6. A. M. Letov, Dinamika Poleta i Upravleniye (Flight Dynamics and Control), Nauka, 1969.
7. Ya. M. Roytenberg, Avtomaticheskoy Upravleniye (Automatic Control), Nauka, 1971.

Computer Simulation of the Dynamics of Complicated Mechanisms of Robot-Manipulators

A. F. VERESHCHAGIN
(Moscow)

A method is suggested for computer simulation of the dynamic behavior of the functional units of robot-manipulators of various constructions, without preliminary derivation of the equations of motion, on the basis of direct implementation of Gauss's principle of least constraint. The mechanism to be simulated is represented as an ensemble of bodies and control motors interconnected kinematically. Recursion relations, describing the total dynamics of the manipulator in compact form, are specially derived for linked mechanisms. Based on the methods developed, a computer program was written to test the efficiency of the control algorithms of manipulators with arbitrary kinematics.

* * *

The design of robot-manipulators requires study of a number of complicated multilink mechanical systems as the control objects. These include mechanical hands, a mechanical human model, etc. Computer simulation of the dynamic behavior of such systems is a powerful method of arriving at a rational kinematic design for the mechanism and efficient control algorithms. The answers to questions arising in the preliminary design stages, in the development, and during tests of robot-manipulators, and also during their use, can often be obtained only by simulation.

The variety of possible constructions and the complexity of the mathematical description of the controlled motion of a multilink mechanism make it critical to develop a system of automatic programming of the algorithms of the mathematical model formulation, and for the simulation of motion in terms of a given kinematic scheme of the mechanism and control law. Moreover, derivation of the extremely cumbersome dynamic equations for a broad class of mechanisms, followed by their programming, leads to the writing of complicated simulation programs requiring considerable internal storage and a high-speed computer.

A method is proposed in this paper for computer simulation of the dynamics of complex mechanisms without deriving the equations of motion, using a direct implementation of Gauss's principle of least constraint. This method makes it possible to write a comparatively simple simulation program, the use of which requires specification of a minimum amount of information about the kinematics of the mechanism and the dynamic characteristics of the individual links.

1. STATEMENT OF THE PROBLEM

The mechanism (functional organ of the robot-manipulator) is considered as an ensemble of bodies (links) and control motors, interconnected in some manner. The connection method is defined by the coupling equations. The velocity of each link can be given by a six-dimensional vector $\dot{x}_i^T = \{\omega_{1i}, \omega_{2i}, \omega_{3i}, v_{1i}, v_{2i}, v_{3i}\}$ $(i = 1, 2, \ldots, n)$, where i is the link number; n is the number of links in the mechanism; $v_i^T = \{v_{1i}, v_{2i}, v_{3i}\}$ is the velocity of the center of mass; $\omega_i = \{\omega_{1i}, \omega_{2i}, \omega_{3i}\}$ is the instantaneous angular velocity of link i; and T denotes the transpose.

We will use the symbol q_j to denote the shaft rotation angle of control motor j $(j = 1, 2, \ldots, p)$.

The possible velocities of the elements of the system being considered for a given configuration satisfy the coupling equations

$$A\dot{x}+B\dot{q}=C, \tag{1.1}$$

where $\dot{x}^T = \{ \dot{x}_1^T, \ldots, \dot{x}_n^T\}$, $\dot{q}^T = \{\dot{q}_1, \ldots, \dot{q}_p\}$; A, B, C are matrices depending on the position coordinates and time, with dimensions $r \times 6n$, $r \times p$, $r \times 1$, respectively; and r is the number of coupling equations.

We divide the active forces on a mechanism with ideal couplings into two classes: external forces, characterized by the principal force vectors $F_i^T = \{F_{1i}, F_{2i}, F_{3i}\}$ and principal torques $M_i^T = \{M_{1i}, M_{2i}, M_{3i}\}$ acting on link $i = 1, 2, \ldots, n$, and the control torques Q_j ($j = 1, 2, \ldots, p$) of the motors. It is convenient to specify the vectors v_i, F_i in a stationary coordinate system, and ω_i, M_i in a moving system with axes coinciding with the principal central inertial axes of link i.

Let us introduce the following notations:

$$W_i = \mathrm{diag}\,\{a_i, b_i, c_i, m_i, m_i, m_i\},\ W = \mathrm{diag}\,\{W_1, \ldots, W_n\},$$

$$\ddot{z}_i^{\mathrm{T}} = \left\{\frac{(b_i - c_i)\,\omega_{2i}\omega_{3i} + M_{1i}}{a_i}, \frac{(c_i - a_i)\,\omega_{3i}\omega_{1i} + M_{2i}}{b_i}, \frac{(a_i - b_i)\,\omega_{1i}\omega_{2i} + M_{3i}}{c_i}, \right.$$

$$\left. \frac{F_{1i}}{m_i}, \frac{F_{2i}}{m_i}, \frac{F_{3i}}{m_i}\right\},$$

$$Q^{\mathrm{T}} = \{Q_1, \ldots, Q_p\},\ \ddot{z}^{\mathrm{T}} = \{\ddot{z}_1^{\mathrm{T}}, \ldots, \ddot{z}_n^{\mathrm{T}}\},$$

where m_i are the masses, and a_i, b_i, c_i are the principal central moments of inertia, of link i; and W_i, W are diagonal matrices with dimensions 6×6 and $6n \times 6n$, respectively.

Assume the position and velocity of the system at time t are given, as well as the forces and torques F_i, M_i ($i = 1, \ldots, n$); Q_j ($j = 1, \ldots, p$). Let us write the equation

$$J = {}^1/_2\,(\ddot{x}-\ddot{z})^{\mathrm{T}} W (\ddot{x}-\ddot{z}) - Q^{\mathrm{T}}\ddot{q} \tag{1.2}$$

and consider those acceleration values $\ddot{x}$, $\ddot{q}$ which are possible for a given configuration and velocity. The possible accelerations of the system satisfy

$$A\ddot{x} + B\ddot{q} = D \quad (D = \dot{C} - \dot{A}\dot{x} - \dot{B}\dot{q}). \tag{1.3}$$

Gauss's principle of least constraint [1] for our mechanical system can be formulated as follows. In the class of all possible accelerations, the true accelerations provide the only minimum in (1.2). Actually, this formula differs from the standard equation for the constraint in Gauss's principle by a quantity that is independent of the accelerations.

When the equations of motion are integrated on a computer by numerical methods, the accelerations for the given coordinates and velocities must be calculated at each step of the integration, once or several times depending on the integration method. In Gauss's principle, determination of the accelerations reduces to the algebraic problem of minimizing the quadratic form (1.2) for the linear restrictions (1.3).

If the inertia of the rotor of motor j, attached to link i, exerts a significant effect on the dynamics of the mechanism, then the term $1/2d_j[\dot{\omega}_i + \varepsilon_j\ddot{q}_j + \dot{q}_j(\omega_i \times \varepsilon_j)]^2 - 1/2d_j(\dot{\omega}_i)^2$ must be added to (1.2), where d_j is the moment of inertia of the rotor with respect to the rotation axis, and ε_j is the unit vector of this axis in the associated coordinate system. In this situation, a_i, b_i, c_i, m_i are calculated for the motor rotor attached to this link.

Thus, to simulate the control motion of a specific mechanism, two nonstandard procedures must be included in the general simulation program operating in accordance with the above-described method. The first of these produces the restriction matrix systems A, B, D of (1.3) from the given configuration and velocity, and the second calculates the control torques Q in accordance with a specific control algorithm. The control algorithm can be formulated from the feedback principle since the torques Q can depend on the position and velocity of the mechanism.

An analytic solution can be written for the minimization problem being considered; however, when a computer is used, iteration methods are advisable because of the large possible dimensions of the matrices involved with a low number density [2]. In addition, when continuous processes are simulated, their characteristics change little from interval to interval. Taking as a first approximation the accelerations of the mechanism which occurred in the previous interval, one can obtain a sufficiently accurate result in the following interval after several iterations.

2. SIMULATION OF THE DYNAMICS OF A MANIPULATOR HAND

In special cases the above simulation scheme can be refined and modified, based on the specific characteristics of the mechanisms being studied. For example, the functional organs of manipulators are generally three-dimensional mechanisms containing several links, connected into a chain with one degree of mobility between the links (see Fig. 1).

Assume the links of the chain are numbered 1, 2, ..., n and the link with number i is connected to link i - 1 by either a cylindrical or telescoping joint, providing, in the first case, rotational relative motion, and in the second case, translational relative motion. An important feature of such mechanisms is the recursive nature of the restrictions (1.1) and (1.3), which in this case have the form

$$\begin{gathered}
\omega_i = L_i L_{i-1}^{\mathrm{T}} \omega_{i-1} + \dot{q}_i e_i (1-\delta_i),\quad v_i = v_{i-1} + L_{i-1}^{\mathrm{T}} (\omega_{i-1} \times r'_{i-1}) - L_i^{\mathrm{T}} (\omega_i \times \rho_i + \dot{q}_i e_i \delta_i), \\
\dot{\omega}_i = L_i L_{i-1}^{\mathrm{T}} \dot{\omega}_{i-1} + (\ddot{q}_i e_i + \dot{q}_i (\omega_i \times e_i)) (1-\delta_i), \\
\dot{v}_i = \dot{v}_{i-1} + L_{i-1}^{\mathrm{T}} (\dot{\omega}_{i-1} \times r'_{i-1} + \omega_{i-1} \times (\omega_{i-1} \times r'_{i-1})) - L_i^{\mathrm{T}} (\dot{\omega}_i \times \rho_i + \omega_i \times (\omega_i \times \rho_i) + \\
+ (\ddot{q}_i e_i + 2\dot{q}_i (\omega_i \times e_i)) \delta_i), \\
\rho_i = r_i'' + q_i e_i \delta_i \quad (i = 1, 2, \ldots, n).
\end{gathered} \tag{2.1}$$

Here $\delta_i = 0$ if the joint is cylindrical, and $\delta_i = 1$ for a telescoping joint; in the first case, q_i is the rotation angle of link i with respect to link i - 1, while in the second case q_i is the relative linear displacement; L_i is the matrix of direction cosines (3 x 3) of the associated coordinate system with axes along the principal inertia axes and with origin at the center of mass of the link; e_i is the unit vector along joint axis i; r_{i-1}', r'' are vectors joining the centers of mass of links i - 1 and i with points on the joint axes that are stationary with respect to the links themselves, and the ends of the vectors r_{i-1}' and ρ_i coincide (see Fig. 2).

After simple manipulations the function (1.2) to be minimized can be represented in the form

$$\begin{gathered}
J = G - \sum_{i=1}^{n} Q_i^* \ddot{q}_i + \cdots = \\
= \sum_{i=1}^{n} \frac{1}{2} m_i (\dot{v}_{1i}^2 + \dot{v}_{2i}^2 + \dot{v}_{3i}^2) + \frac{1}{2} (a_i \dot{\omega}_{1i}^2 + b_i \dot{\omega}_{2i}^2 + c_i \dot{\omega}_{3i}^2) - \\
- (b_i - c_i) \omega_{2i} \omega_{3i} \dot{\omega}_{1i} - (c_i - a_i) \omega_{3i} \omega_{1i} \dot{\omega}_{2i} - (a_i - b_i) \omega_{1i} \omega_{2i} \dot{\omega}_{3i} + \cdots - Q_i^* \ddot{q}_i + \cdots,
\end{gathered}$$

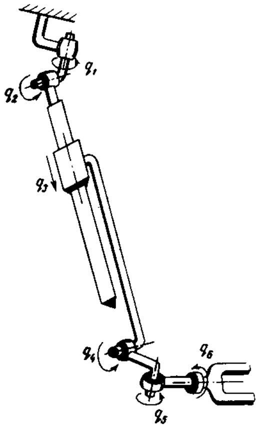

Fig. 1. Typical design of functional organ of a manipulator.

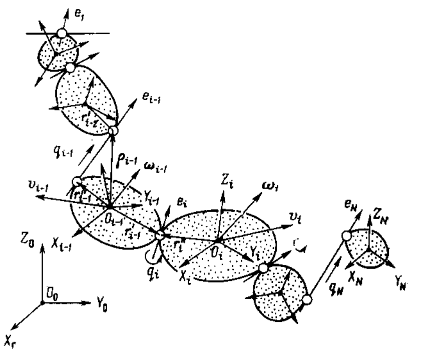

Fig. 2. Kinematic scheme of an open mechanism.

where Q_i* are the generalized forces created by the external forces and the torques of the control motors, and G is the Gibbs-Appel function (the acceleration energy). The terms which do not depend on the accelerations, and consequently do not affect the result of the minimization, are not included in the last formulas

The recursion relations (2.1) make it possible to calculate the value of G directly as a function of q, $\dot{q}$, $\ddot{q}$. Based on a given motion of the base of the manipulator hand $\{\omega_0, v_0, \omega_0, \dot{v}_0\}$, we determine from (2.1) the values of the vectors ω_i, v_i, $\dot{\omega}_i$, $\dot{v}_i$ successively for i = 1, 2, ..., n; the components of these vectors enter into the acceleration energy formula.

In one version of the simulation program for the dynamics of the manipulator control motion, implemented on a BESM-4 computer, we made use of the ability to compute the Gibbs-Appel function inthe following manner. This function is quadratic with respect to $\ddot{q}_1$, $\ddot{q}_2$, ..., $\ddot{q}_n$: $G = \frac{1}{2}\sum_{\substack{i=1\\j=1}}^{n} \alpha_{ij}\ddot{q}_i\ddot{q}_j + \sum_{i=1}^{n} \beta_i\ddot{q}_i + \gamma$, and it yields the equations of motion of the mechanical system in the simplest form (Gibbs-Appel form): $\partial G/\partial \ddot{q}_i = Q_i^*$ (i = 1, 2, ..., n), or

$$\sum_{j=1}^{n} \alpha_{ij}\ddot{q}_i + \beta_i = Q_i^* \quad (i = 1, 2, \ldots, n). \tag{2.2}$$

To calculate G the acceleration vector $\ddot{q}$ must also be specified, in addition to the coordinate q and velocity $\dot{q}$. The first two vectors q and $\dot{q}$ define the actual instantaneous position and motion of the mechanism, and the components of the vector $\ddot{q}$ in the program are selected with the goal of rapid identification of the instantaneous values of the elements of the symmetric matrix $\alpha = [\alpha_{ij}]$, the vector $\beta = \{\beta_1, \ldots, \beta_n\}$, and the quantity γ.

If, for example, the three values $\ddot{q}_0 = \{0, \ldots, 0\}$, $\ddot{q}_+ = \{0, \ldots, 0, 1, 0, \ldots, 0\}$, $\ddot{q}_- = \{0, \ldots, \ldots, 0, -1, 0, \ldots, 0\}$, where the unity stands in place i, correspond to three acceleration energies G_0, G_+, and G_-, then $\alpha_{ii} = G_+ + G_- - 2G_0$, $\beta_i = (G_+ - G_-)/2$. The nondiagonal elements of the matrix are calculated similarly.

Formulation of the equations of motion in the form (2.2) requires $1 + n + n(n + 1)/2$ applications of the procedure for calculating the Gibbs-Appel function, for different specially chosen values of the acceleration vector. Individual blocks of the procedure are also used for calculating the generalized force components created by the force of gravity and by other external forces.

After identification of the elements of the matrix α and vector β we have to return to the non-standard procedure, implementing the specific control algorithm, which finally establishes the instantaneous values of all the generalized forces. The calculated accelerations are integrated by the Runge-Kutta method or some other method.

The corresponding block diagram of the simulation is given in Fig. 3.

With this approach, the simulation program has minimum dimensions and is extremely simple to implement; however, the computer time expended in the simulation increases quadratically with an increase in the number of links in the mechanism. In practice, this program can be used on BESM-4 and M-220 computers for mechanisms with 5-7 links. Therefore, in order to speed up the simulation process, the problem of minimizing the extent of the constraint is solved below by means of dynamic programming, making it possible to represent all necessary algorithms in a recursive form convenient for computer use.

3. DYNAMIC PROGRAMMING AND THE MANIPULATOR DYNAMICS

Using the previous notations, let us represent the acceleration relations (2.1) in matrix form as

$$\ddot{x}_i = A_i\ddot{x}_{i-1} + B_i\ddot{q}_i + f_i \quad (i=1, 2, \ldots, n), \tag{3.1}$$

where A_i, B_i, f_i are matrices that depend on the system position and velocity, and have dimensions of 6 × 6, 6 × 1, 6 × 1, respectively. The vector $\ddot{x}_0$ is given if the column motion is given (if the mechanism is attached to a stationary column, then $\dot{x}_0 = \ddot{x}_0 = 0$). Here it is also easy to take into consideration the case when the column is not kinematically connected to any inertial coordinate system and, like

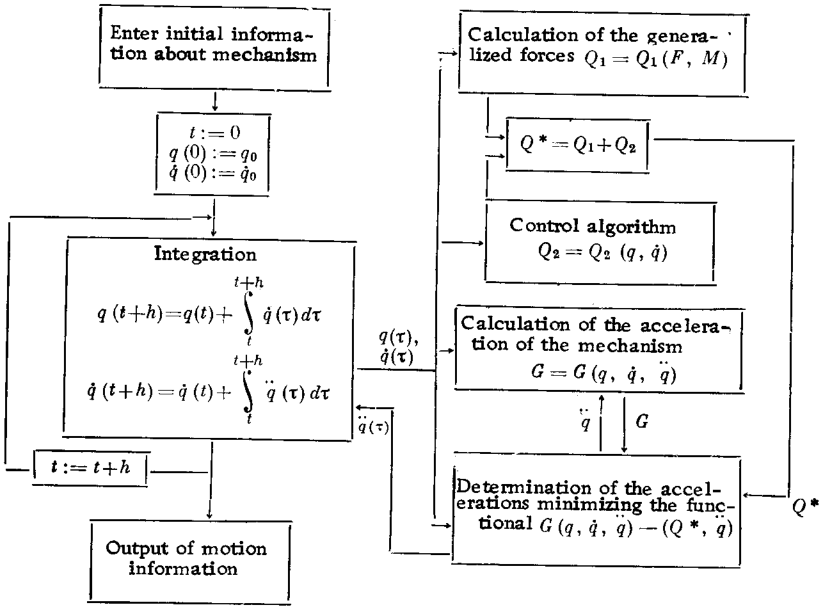
Enter initial information about mechanism
$t := 0$
$q(0) := q_0$
$\dot{q}(0) := \dot{q}_0$
Integration
$q(t+h)=q(t)+\int_t^{t+h}\dot{q}(\tau)d\tau$
$\dot{q}(t+h)=\dot{q}(t)+\int_t^{t+h}\ddot{q}(\tau)d\tau$
$t := t+h$
Output of motion information
$q(\tau)$, $\dot{q}(\tau)$
$\ddot{q}(\tau)$
Calculation of the generalized forces $Q_1 = Q_1(F, M)$
$Q^* = Q_1+Q_2$
Control algorithm $Q_2 = Q_2(q, \dot{q})$
Calculation of the acceleration of the mechanism $G = G(q, \dot{q}, \ddot{q})$
$\ddot{q}$
G
Determination of the accelerations minimizing the functional $G(q, \dot{q}, \ddot{q}) - (Q^*, \ddot{q})$
Q^*

Fig. 3

the links, has an inertia characterized by the matrix W_0. Then the vector $\ddot{x}_0$ must be determined.

Let us represent the function (1.2) to be minimized in additive form as

$$J=\sum_{i=1}^{n}\frac{1}{2}(\ddot{x}_i-\ddot{z}_i)^{T}W_i(\ddot{x}_i-\ddot{z}_i)-Q_i\ddot{q}_i. \tag{3.2}$$

To sum up, we have a multistep dynamic programming problem for determining $\ddot{q}_1, \ddot{q}_2, \ldots, \ddot{q}_n$ [3]. We determine the minimal function

$$I_{k-1}(\ddot{x}_{k-1})=\min_{\ddot{q}_j}\left\{\sum_{j=k-1}^{n}\frac{1}{2}(\ddot{x}_j-\ddot{z}_j)^{T}W_j(\ddot{x}_j-\ddot{z}_j)-\sum_{j=k}^{n}Q_j\ddot{q}_j\right\}$$
$$(j=k, k+1, \ldots, n),$$

for which the basic dynamic programming recursion relation

$$I_{k-1}(\ddot{x}_{k-1})=\min_{\ddot{q}_k}\{{}^1/_2(\ddot{x}_{k-1}-\ddot{z}_{k-1})^{T}W_{k-1}(\ddot{x}_{k-1}-\ddot{z}_{k-1})-Q_k\ddot{q}_k+ + I_k(A_k\ddot{x}_{k-1}+B_k\ddot{q}_k+f_k)\}, \tag{3.3}$$

is valid, and $\ddot{q}_k$ is determined from the necessary condition that the right side of this formula be a minimum.

The solution of the basic equation can be represented in the form

$$I_k(\ddot{x}_k)={}^1/_2\ddot{x}_k^{T}P_k\ddot{x}_k+R_k^{T}\ddot{x}_k+S_k \quad (k=0, 1, \ldots, n), \tag{3.4}$$

where P_k is a symmetric 6×6 matrix, R_k is a 6×1 vector, and S_k is a scalar quantity. Substitution of this function into (3.3), taking into consideration the necessary condition for a minimum, and equating the coefficients of like powers of the components of the vector $\ddot{x}_{k-1}$ make it possible to obtain a solution by means of the following recursion formulas:

$$\ddot{q}_i = (Q_i - ((\ddot{x}_{i-1}{}^T A_i{}^T + f_i{}^T) P_i + R_i{}^T) B_i)(B_i{}^T P_i B_i)^{-1}, \tag{3.5}$$

$$P_{i-1} = W_{i-1} + A_i{}^T (P_i - P_i B_i (B_i{}^T P_i B_i)^{-1} B_i{}^T P_i) A_i, \tag{3.5a}$$

$$\begin{gathered} R_{i-1}{}^T = -\ddot{z}_{i-1}{}^T W_{i-1} + (R_i{}^T + f_i{}^T P_i) A_i + (Q_i - (R_i{}^T + f_i{}^T P_i) B_i) \times \\ \times (B_i{}^T P_i B_i)^{-1} B_i{}^T P_i A_i, \\ P_n = W_n,\ R_n{}^T = -\ddot{z}_n{}^T W_n \ (i = 1, 2, \ldots, n). \end{gathered} \tag{3.5b}$$

The solution yields the equations of motion of the mechanism, written in algorithmic form and solved for the accelerations, and the solution is achieved without formal use of matrix inversion [$(B_i{}^T P_i B_i)^{-1}$ is a scalar quantity]. The amount of work involved in calculations with the stated recursion formulas increases linearly with an increase in the number of links in the mechanism. Therefore relations (3.5)-(3.5b) make it possible to write an efficient program to simulate the dynamic behavior of a multilink manipulator hand in the control process. In this program the acceleration calculation algorithm for given mechanism configuration and velocity consists of the following steps.

1. Calculation of the matrices P_i successively for $i = n,\ n - 1, \ldots$ by means of (3.5a).

2. Calculation of the control torques Q_i ($i = 1, 2, \ldots$) of the motors by means of the procedure that implements the specific control algorithm.

3. Calculation of the vectors R_i successively for $i = n,\ n - 1, \ldots$ by means of (3.5b).

4. Calculation of the accelerations of the relative displacements $\ddot{q}_i$ and the accelerations of the links $\ddot{x}_i$ ($i = 1, 2, \ldots, n$) by means of (3.5) and (3.1). If no restrictions are imposed on the value of $\ddot{x}_0$, i.e., $\ddot{x}_0$ is also an argument that minimizes (3.2), then the extremum value, as follows from (3.4), is determined from $\ddot{x}_0 = -P_0{}^{-1} R_0$.

The recursion relations on which these steps are based are easily programmed for a computer, and make it possible to increase the simulation speed by a factor of 10-12 compared with the above-described program based on calculating the acceleration energy.

Based on the method discussed, we wrote and successfully used an ALGOL program for automatic formulation and integration of the equations of the controlled motion of a multilink manipulator with an arbitrary kinematic design.

The program is used for studying and testing the functioning of the control algorithms being developed.

The method of simulation by direct implementation of Gauss's least constraint principle can serve as the basis for a systematic approach to programming the motions of complicated mechanisms, eliminating the necessity of writing special programs for each special problem.

Submitted June 4, 1974

REFERENCES

1. L. Pars, Analiticheskaya Dinamika (Analytical Dynamics), Nauka, 1971.
2. N. S. Bakhvalov, Chislennyye Metody (Numerical Methods), Vol. 1, Nauka, 1973.
3. R. Bellman, Dynamic Programming, Princeton Univ. Press, (in Russian transl.), Foreign Literature Press, 1960.

□◆◆◆□