A Symbolic Formulation of Dynamic Equations For a Manipulator With Rigid and Flexible Links

J. Lin and F.L. Lewis

The International Journal of Robotics Research 1994 13: 454

DOI: 10.1177/027836499401300506

Communication

J. Lin
F. L. Lewis
Automation and Robotics Research Institute
The University of Texas Arlington
Fort Worth, Texas 76118

A Symbolic Formulation of Dynamic Equations For a Manipulator With Rigid and Flexible Links

Abstract

The objective of this article is to present an efficient procedure for computer generation of the dynamic equations for a planar robot manipulator with arbitrarily assigned rigid or flexible links using any desired flexible mode shape functions. The dynamic equations for the serial link manipulator are derived using Lagrange's formulation and elastic deflection with the assumed-mode method. Fewer approximations are made than in other approaches, resulting in greater accuracy. A method to determine the centrifugal and Coriolis matrix is presented that yields an important structural property. The approach is systematic and allows a symbolic program to be written in Mathematica using a system of several groups and a constructed database. Four examples are illustrated to verify the dynamic equations. The stability of the zero dynamics is compared for different mode shape functions.

1. Introduction

Control of mechanical manipulators by tracking a desired trajectory is an extremely important problem. The problem becomes more complex when the robot possesses flexibility or multiple links. A prerequisite for developing the necessary control algorithm is an efficient method for determining the dynamic equations of the manipulator. Traditionally, robotics engineers have derived the dynamic equations of flexible link arms by hand computations. Because manual symbolic expansion of the robot dynamic equations is tedious, time consuming, prone to errors, and requires previous experience, an automated derivation process is highly desirable. This procedure should allow the use of any specified mode shape functions.

The International Journal of Robotics Research,
Vol. 13, No. 5, October 1994, pp. 454–466,

Although a variety of schemes to formulate the dynamic equations in a symbolic form for robots with rigid links have been studied by many researchers (Faessler 1986; Hollerbach 1980; Leu and Hemati 1986), an efficient algorithm and computer program that generates the dynamical equations for a robot with arbitrarily assigned rigid/flexible links is not yet available. In recent years, some researchers have concentrated on the symbolic formulation of dynamic equations for robot manipulators with elastic links. These (Low 1987; Low and Vidyasagar 1988) present a procedure for deriving dynamic equations for manipulators containing both rigid and flexible links. The equations are derived using Hamilton's principle and are nonlinear integro-differential equations. In these articles, the formulation is based on expressing the kinetic and potential energies of generalized coordinates. In the case of flexible links, the mass distribution and flexibility must be taken into account. However, the integro-differential equation form of the final dynamic equation is not suitable for computer simulation and controller design. Furthermore, the derivation process used in these articles makes use of the variation principle and a substantial amount of manipulation, making the procedure tedious and complex.

An explicit, nonrecursive symbolic form of the dynamic model for robotic manipulators with compliant links and joints is developed in Centinkunt and Book (1989), with the final form of the equations having a form similar to the rigid manipulator equation. Furthermore, Centinkunt and Ittop (1992) developed a program in Reduce to automate the symbolic modeling of the dynamics of robotic manipulators with flexible links. In the case of flexible arms, a point along the beam can be described in a fixed reference coordinate system by two transformations between the coordinate systems. The dynamics computations are performed in the link coordinate

system, as is also done in Li and Sankar (1992), where the kinematic information is computed using a forward recursion from the base to the hand tip, and the dynamic information is computed using a return recursion. However, in the derivation procedure of those papers, sparse transformation matrices arise owing to the combination of translation with rotation, but the sparseness is not taken advantage of, resulting in computational inefficiency.

A symbolic program in Macsyma for a serial four-link robot with a two-revolute joint angle and alternating flexible and rigid links (FRFR) is shown in Lee (1992). The derivation procedure is similar to that in Centinkunt and Book (1989) and Yuan (1989), and the program is structured specifically for the special case of FRFR, with no provision for varying the number of links or the flexible/rigid link order, or an option to analyze only flexible or only rigid link arms. The dynamic equations are derived for a class of multilink manipulators with only the last link flexible as shown in Wang and Vidyasagar (1992). In this article, some assumptions that simplify the dynamic equations for control purposes are described, so that the derivation applies only in restricted cases.

Closed-form equations of motion for planar lightweight robot arms with multiple flexible links are presented in De Luca and Siciliano (1991). The kinematic model is based on standard frame transformation matrices describing both rigid rotation and flexible displacement under a small deflection assumption. Links are modeled as Bernoulli-Euler beams with proper clamped-mass boundary conditions. The aim of this work is limited to the case of planar manipulators.

In addition to each of the drawbacks specific to each article, those articles have some additional limitations: no clearly defined centrifugal and Coriolis matrix with desirable (e.g., skew-symmetric) properties and no capability to arbitrarily handle assigned rigid and flexible links.

The objective of this article is to present an efficient procedure for computer generation of symbolic modeling equations for planar robot manipulators. This method allows any link to be arbitrarily assigned as a rigid or a flexible link, with the option of adding a payload at the tip of the last link. It does not make some common approximations used in other work, and so yields more accurate dynamics than many other techniques (see example 4.2). The symbolic form of equations that describes the serial link manipulator dynamic equations is developed using Lagrange's equations and elastic deflection with the assumed-mode method. The computer automated symbolic expansion program is written in Mathematica symbolic software. The concepts in this paper can be extended to nonplanar arms in R^3. In fact, example 4.3 shows that in special cases it applies to 3D arms as it is.

The outline of the article is as follows. In Section 2, an algorithm for deriving the dynamic equation is developed. A procedure is given for determining the centrifugal and Coriolis matrix that satisfies the important skew-symmetric property. In Section 3, the computer symbolic program is presented, and its advantages are also discussed. In Section 4, four examples are given to illustrate the procedure. The effects of higher modes on the inertia matrix and system dynamic characteristics are investigated and discussed. The stability of the zero dynamics is compared for different mode shape functions. Section 5 includes some concluding remarks and a discussion of future work.

2. General Algorithm for Flexible/Rigid Link Dynamics

In this section, an efficient and accurate algorithm is developed for modeling the dynamic equations of a planar lightweight arm. The algorithm generates the kinematic description in moving coordinates, using the rotation matrix from flexible frame to rigid frame to express position vectors in the base frame. It uses Lagrangian dynamics and the assumed-mode approach to derive the equations of motion. Although this article treats 2D planar arms, the approach generalizes to arbitrary arms in R^3. In fact, the algorithm applies if the arm is planar with the exception of the last link, which can have motion out of plane (example 4.3). The algorithm relies on fewer approximations than other approaches and so yields greater accuracy (see example 4.2).

2.1. *System Description*

Consider the open-chain 3D kinematic structure shown in Figure 1, which represents a robotic manipulator with serial links connected by revolute joints. $O(x_0y_0z_0)$ is the base frame, θ_r are joint angles at the $r-1$ joint, and mp is the tip payload mass on the last link. Assume the y_r coordinates are always attached along the link axial direction on the rigid body moving frame associated to link r. The elements of the manipulator are numbered, and moving coordinates are assigned as shown. Define n as the number of rigid links, m as the number of flexible links, T as the total number of links, and gen as the total numbers of generalized coordinates, $q = \{q_R, q_F\}$. In general, any link can be either flexible or rigid. Notation is as follows:

The local position vector for the center of mass for rigid link i is $h_c[i]$.

The local position vector for rigid link i is $h_R[i]$.

The local position vectors for any point on flexible link j is $h_F[j]$.

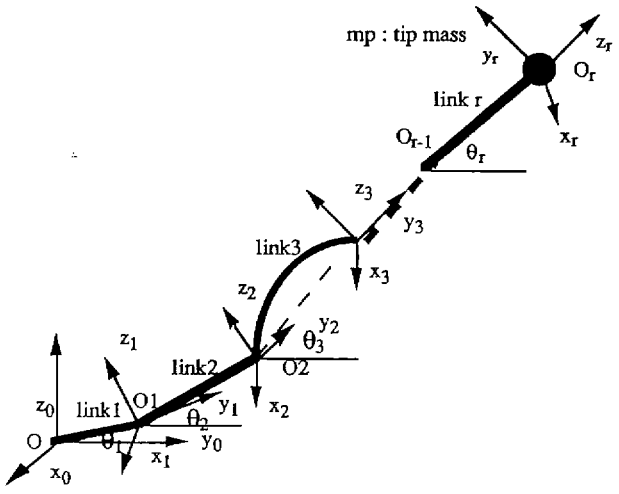

Fig. 1. Kinematic description of a serial link (rigid plus flexible).

Each of the flexible links is regarded as a Bernoulli-Euler beam, the elastic deflection is approximated by the series composed of a linear combination of mode shape functions multiplied by time-dependent generalized coordinates. Therefore,

$$h_F[j] = [\Delta x_j \; y + \Delta y_i \; \Delta z_j]^T;$$

where Δx, Δy, Δz are the link displacements along the x_r, y_r, z_r axis.

$$\begin{aligned}
\Delta x_j(y_j,t) &= \sum_{k=1}^{\infty} \phi_{xjk}(y_j) U_{jk}(t), \Delta y_j(y_j,t) \\
\Delta y_j(y_j,t) &= \sum_{k=1}^{\infty} \phi_{yjk}(y_j) W_{jk}(t), \Delta z_j(y_j,t) \qquad (1) \\
\Delta z_j(y_j,t) &= \sum_{k=1}^{\infty} \phi_{zjk}(y_j) V_{jk}(t)
\end{aligned}$$

Where j represents the link number, k represents the mode number, and $0 \leq y_j \leq l_j$.

U_{jk}, W_{jk}, V_{jk} are flexible mode generalized coordinates, and ϕ_{xjk}, ϕ_{yjk}, ϕ_{zjk} are mode shape functions that are dependent on the boundary-value problem (i.e., pinned-pinned, clamped-free, clamped-loaded, etc.).

The usual formulation of a flexible manipulator carrying a load is derived by assuming a free boundary condition at the end of the link. This method is usually used due to the difficulty in accounting for time-varying and/or unknown masses and inertias. To achieve a more exact model, Gorman (1975) introduced some analytical forms of the nonclassic boundary conditions for a Bernoulli-Euler beam. We use this approach, as it is more suitable for considering the mass boundary conditions in the partial differential equation. While these new boundary conditions will cause the partial differential equation to become more complicated, the equation can be solved symbolically with appropriate software.

2.2. *Rotation Matrix and Position Vector*

The transformation and rotation matrix will now be applied to find the position vector of each link. A_r is the homogeneous transformation matrix from the r coordinate system to the $r-1$ coordinate system and is of the form (Spong and Vidyasagar 1989)

$$A_r = \begin{bmatrix} R_{r-1}^r & d_{r-1}^r \\ 0 & 1 \end{bmatrix} \qquad (2)$$

Hence

$$T_0^r = A_1 \cdots A_r = \begin{bmatrix} R_0^r & d_0^r \\ 0 & 1 \end{bmatrix} \qquad (3)$$

The matrix R_{r-1}^r expresses the orientation of frame r relative to frame $r-1$ and is given by the rotational parts of the A_r matrices as $R_0^r = R_0^1 \cdots R_{r-1}^r$. The vectors d_0^r are given recursively by the formula

$$d_0^r = d_0^{r-1} + R_0^{r-1} d_{r-1}^r. \qquad (4)$$

If the system contains all rigid links, the joint angles are easy and simple to define, since there is no elastic deflection at the link. Otherwise, for the flexible members the joint angles will need to be modified because of elastic deflection. The two cases shown below cover all possibilities and are based on a well-known geometric concept that defines the joint angles between flexible and rigid links.

CASE 1. Rigid-rigid link or rigid-flexible link sequences.

The equivalent joint angle is θ_r^*. For case 1, $\theta_r^* = \theta_r$, because there is no elastic deflection at the end of the $r-1$ link (Fig. 2).

CASE 2. Flexible-rigid link or flexible-flexible link sequences.

Assuming the elastic deflection is small, this implies $\Delta\theta_r$ is small and

$$\tan(\Delta\theta_r) = \Delta\theta_r = \frac{\Delta z(L_{r-1})}{L_{r-1}},$$

owing to elastic deflection $\Delta z(L_{r-1})$ at the end of $r-1$ link (Fig. 3). Furthermore, the equivalent joint angle θ_r^* will be defined as $\theta_r^* = \theta_r + \Delta\theta_r$. Because the deflection is approximated by the series of mode shape functions multiplied by time-dependent generalized coordinates, $\Delta z(L_{r-1})$ can be indicated as

$$\sum_{k=1}^{\infty} \phi_{zjk}(L_{r-1}) V_{jk}(t).$$

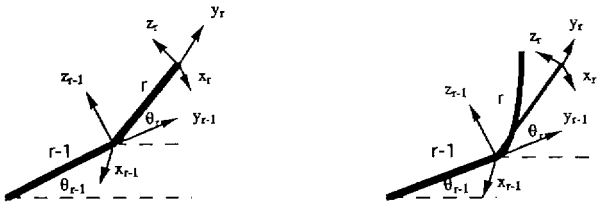

Fig. 2. Kinematic description for rigid-rigid and rigid-flexible link sequences.

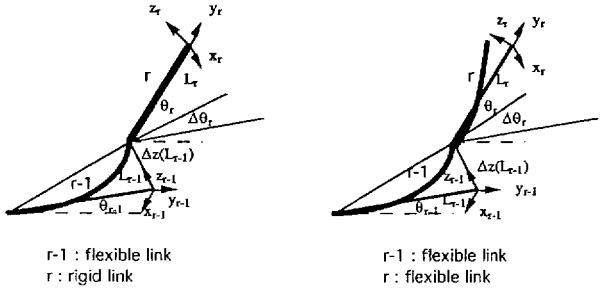

Fig. 3. Kinematic description for flexible-rigid and flexible-flexible link sequences.

The equivalent joint angle concept is for the planar case, but it can also be extended to more degrees of freedom in elastic deflection (e.g., the general nonplanar arm). It also applies as given here if only the last link is nonplanar (e.g., example 4.3). It is easy to implement the rotation matrix efficiently in a symbolic program.

A reduction in computations associated with the transformation matrix can be obtained through a reformulation of the Lagrangian dynamics in terms of a 3×3 rotational matrix rather than the 4×4 homogeneous rotation-translation matrices used in Cetinkunt and Book (1989), Cetinkunt and Ittop (1992), Lee (1992), and Yuan (1989). Although more convenient in setting up the dynamics, with 4×4 matrices inefficiencies arise due to sparseness and the combination of translation with rotation (Hollerbach 1980). The 3×3 formulation takes advantage of sparseness to simplify computations.

Therefore, we take the position vector for the center of mass of a rigid link i in terms of the base frame as

$$P_{ci} = PP_{r-1} + R_0^r h_{ci} = PP_{r-1} + R_0^1 R_1^2 R_2^3 \cdots R_{r-1}^r h_{ci}. \tag{5}$$

Similarly, the position vector for any point on a flexible link j can be defined in terms of the base frame by

$$P_j = PP_{r-1} + R_0^r h_j = PP_{r-1} + R_0^1 R_1^2 R_2^3 \cdots R_{r-1}^r h_j, \tag{6}$$

where r = the link number of the system and PP_{r-1} = the position vector from the base to the $r-1$ joint. The angular velocity for the rigid link is

$$\begin{aligned} \omega_i &= R_{r-1,r}\omega_{i-1} + \dot{\theta}_i \hat{X} \\ &= R_{r-1}^r \cdots R_0^1 \omega_0 + R_{r-1}^r \cdots R_1^2 \dot{\theta}_i \hat{X} + \cdots + \dot{\theta}_i \hat{X} \end{aligned} \tag{7}$$

The tip mass on the last link T in terms of the base frame is $P_T(L_T)$. In eqs. (5)–(7), the ith rigid link (or the jth flexible link) is the rth link of the whole system.

Differentiation of the position vector eq. (5) and (6) allows the velocity of the rigid link and the flexible link to be expressed as $\dot{P}_{ci}$ and $\dot{P}_j$. Hence, the tip mass velocity is $\dot{P}_T(L_T)$.

2.3. Kinetic and Potential Energy

The kinetic energy of the rigid link is the sum of two terms: the translation energy obtained from the entire mass of the link at the center of mass, and the rotational energy of the body about the center of mass.

$$K = \frac{1}{2} m_i \dot{P}_{ci}^T \dot{P}_{ci} + \frac{1}{2} \omega_i^T I_i \omega_i, \tag{8}$$

where m_i = the mass for each rigid link and I = the inertia rotational mass for each rigid link.

The kinetic energy for each elastic link in the system is

$$K = \frac{1}{2} \int_{\text{body}} \dot{P}_j^T \dot{P}_j \, dm = \frac{1}{2} \int_0^{l_j} \rho A \dot{P}_j^T \dot{P}_j \, dy. \tag{9}$$

Note that y is the distance along the flexible link. The kinetic energy of the tip mass on the last link is

$$K = \frac{1}{2} m_p \dot{P}_T^T(L_T) \dot{P}_T(L_T). \tag{10}$$

The total kinetic energy of the system equals the sum of the kinetic energy for each rigid link, plus each flexible link and the tip mass. In eq. (11), J_{ci} is the Jacobian matrix mapping the coordinate velocities $\dot{q}$ to the ith rigid link. J_{ai} is the Jacobian matrix mapping $\dot{q}$ to the angular velocity of the ith rigid link, J_j is the Jacobian mapping the $\dot{q}$ to the velocity of any point in the jth flexible link, and J_p is the Jacobian mapping $\dot{q}$ to the velocity of the tip mass. Expressing velocity in terms of Jacobian matrices, the kinetic energy will become

$$\begin{aligned} K &= \frac{1}{2} \sum_{i+1}^{n} \dot{q}^T (m_i J_{ci}^T J_{ci} + J_{ai}^T I_i J_{ai}) \dot{q} \\ &\quad + \frac{1}{2} \sum_{j=1}^{m} \int_0^{l_j} \rho_j A_j (\dot{q}^T J_j^T J_j \dot{q}) \, dy \\ &\quad + \frac{1}{2} m_p \dot{q}^T J_p^T J_p \dot{q} \\ &= \frac{1}{2} \sum_{i=1}^{n} \dot{q}^T H_i \dot{q} + \frac{1}{2} \sum_{j=1}^{m} \int_0^{l_j} \dot{q}^T \hat{H}_j \dot{q} \, dy \\ &\quad + \frac{1}{2} \dot{q}^T \bar{H} \dot{q}, \end{aligned} \tag{11}$$

where

$$H_i = m_i J_{ci}^T J_{ci} + J_{ai}^T I_i J_{ai}, \tag{12}$$

$$\hat{H}_j = \rho_j A_j J_j^T J_j, \tag{13}$$

$$\tilde{H} = m_p J_p^T J_p, \tag{14}$$

H_i, $\hat{H}_j$, and $\tilde{H}$ are all symmetric matrices.

The potential energy of the links is composed of two parts: the gravitational energy, and the strain potential energy owing to the flexure of the links. The potential energy in terms of the base frame is

$$V = V_g + V_s = \text{gravity energy} + \text{strain energy}$$

$$V_g = \sum_{i=1}^{n} m_i g P_{ci} + \sum_{j=1}^{m} \int_0^{l_j} \rho_j A_j g P_j \, dy + m_p g P_T(L_T) \tag{15}$$

$$V_s = \sum_{i=1}^{m} \frac{1}{2} \int_0^{l_j} \left[EI_x \left(\frac{\partial^2(\Delta x)}{\partial y^2} \right)^2 + EI_z \left(\frac{\partial^2(\Delta z)}{\partial y^2} \right)^2 + EA \left(\frac{\partial \Delta y}{\partial y} \right)^2 + GJ \left(\frac{\partial \phi y}{\partial y} \right)^2 \right] dy. \tag{16}$$

where EI_x and EI_x are the bending stiffnesses in the OX and OZ directions, respectively; EA is the extensional stiffness; and GJ is the torsional stiffness. For greater accuracy, we include the last two terms in (16), which are usually neglected.

2.4. Equations of Motion

Now, we derive the equations of motion using Lagrange's formulation:

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{q}}\right) - \left(\frac{\partial K}{\partial q}\right) + \left(\frac{\partial V}{\partial q}\right) = Q \qquad q = \{q_R, q_F\} \tag{17}$$

The gravity vector has components of

$$G_\alpha = \frac{\partial V_g}{\partial q_\alpha} = \sum_{i=1}^{n} m_i g J_{ci} + \sum_{j=1}^{m} \int_0^{l_j} \rho_j A_j g J_j \, dy + m_p g J_p. \tag{18}$$

The same approach for the strain energy shows

$$U_\alpha = \frac{\partial V_s}{\partial q_\alpha}. \tag{19}$$

After defining all the terms we are able to write the dynamic equations as

$$\sum_{\beta=1}^{\text{gen}} M_{\alpha\beta}(q)\ddot{q}_\beta + \sum_{\gamma=1}^{\text{gen}} \sum_{\beta=1}^{\text{gen}} D_{\alpha\beta\gamma}\dot{q}_\gamma \dot{q}_\beta + \frac{\partial V}{\partial q_\alpha} = Q_\alpha; \quad \alpha = 1, \cdots, \text{gen}, \tag{20}$$

where

$$M_{\alpha\beta} = \left(\sum_{i=1}^{n} H_{\alpha\beta i} + \sum_{j=1}^{m} \int_0^{l_j} \hat{H}_{\alpha\beta j} \, dy + \tilde{H}_{\alpha\beta} \right) \tag{21}$$

$$D_{\alpha\beta\gamma} = \left(\sum_{i=1}^{n} h_{\alpha\beta\gamma i} + \sum_{j=1}^{m} \int_0^{l_j} \hat{h}_{\alpha\beta\gamma j} \, dy + \tilde{h}_{\alpha\beta\gamma} \right) \tag{22}$$

and

$$h_{\alpha\beta\gamma i} = \left(\frac{\partial H_{\alpha\gamma i}}{\partial q_\beta} - \frac{1}{2} \frac{\partial H_{\beta\gamma i}}{\partial q_\alpha} \right)$$

$$\hat{h}_{\alpha\beta\gamma j} = \left(\frac{\partial \hat{H}_{\alpha\gamma j}}{\partial q_\beta} - \frac{1}{2} \frac{\partial \hat{H}_{\beta\gamma j}}{\partial q_\alpha} \right)$$

$$\tilde{h}_{\alpha\beta\gamma} = \left(\frac{\partial \tilde{H}_{\alpha\gamma}}{\partial q_\beta} - \frac{1}{2} \frac{\partial \tilde{H}_{\beta\gamma}}{\partial q_\alpha} \right)$$

We denote by $H_{\alpha\gamma i}$, and so on, the (α, γ) component of matrix H_i.

The generalized force is derived from the virtual work as a result of the joint torque

$$\delta W = \sum_{\alpha=1}^{\text{gen}} Q_\alpha \delta q_\alpha = \tau \sum_{\alpha=1}^{\text{gen}} \phi'_\alpha(0) \delta q_\alpha,$$

where $Q_\alpha = \tau$, when α is the rigid mode, $Q_\alpha = \phi'_\alpha(0)\tau$ when α is the flexible mode, and τ is the input torque at the base motor. Therefore, the generalized force vector Q can be defined as $Q = B \cdot \tau$, where :

$$B^T = [I \phi'_1(0) \cdots \phi'_s(0)];$$

s indicates the number of modes in the elastic link, and τ is the input torque at the base of the link, B is the input matrix, which depends on the clamped link assumptions. Note that in the pinned-pinned mode shape function, $\phi'_s(0) \neq 0$, and in the clamped-free mode shape function, $\phi'_s(0) = 0$.

The components of the dynamic model should be explicitly separated into matrix form to exhibit the inertia (M), centrifugal and Coriolis (D), structural stiffness (U), and gravitational (G) terms separately. The next result is the basis for this article.

THEOREM 1. The dynamics of the planar rigid/flexible link arm are given by

$$[M(q)]\ddot{q} + [D(q, \dot{q})]\dot{q} + [U]q + G(q) = Q, \tag{23}$$

with $M(q)$ having elements $M_{\alpha\kappa}$ given by (21), and $D(q, \dot{q})$ having elements $D_{\alpha\kappa}$ defined by

$$D_{\alpha\kappa} = \sum_{\gamma=1}^{\text{gen}} D_{\alpha\kappa\gamma}(q)\dot{q}_\gamma \tag{24}$$

with $D_{\alpha\kappa\gamma}$ given in (22).

Moreover

$$M_{\alpha\kappa} = \frac{\partial Q_\alpha}{\partial \ddot{q}_\kappa} \tag{25}$$

$$D_{\alpha\kappa} = \frac{1}{2}\frac{\partial Q_\alpha}{\partial \dot{q}_\kappa} \tag{26}$$

$$\alpha, \kappa = 1, \cdots, \text{gen.}$$

Finally, $\dot{M} - 2D$ is skew symmetric.

Remark. This theorem provides a reliable and direct way (e.g., (25), (26)) to find $M(q)$ and $D(q, \dot{q})$. Although the definition of $D(q, \dot{q})$ is not unique, (26) yields the $D(q, \dot{q})$ that makes $\dot{M} - 2D$ skew symmetric. This relation is fundamental in the design of adaptive or robust controls for serial link manipulators.

Proof of Theorem 1: The dynamics have been derived, so it remains only to show (25), (26), and the skew-symmetric property.

(i) Because none of the terms, except the first terms in Q_α, are functions of $\ddot{q}_\kappa$, then

$$\frac{\partial Q_\alpha}{\partial \ddot{q}_\kappa} = M_{\alpha\kappa}; \qquad \alpha, \kappa = 1, \cdots, \text{gen.}$$

(ii)

$$\frac{\partial Q_\alpha}{\partial \dot{q}_\kappa} = \frac{\partial}{\partial \dot{q}_\kappa}\left[\sum_{\gamma=1}^{\text{gen}}\left(\sum_{\beta=1}^{\text{gen}}\left(\sum_{i=1}^{n}\frac{\partial H_{\alpha\gamma i}}{\partial q_\beta} + \sum_{j=1}^{m}\int_0^{l_j}\frac{\partial \hat{H}_{\alpha\gamma j}}{\partial q_\beta}\,dy + \frac{\partial \tilde{H}_{\alpha\gamma}}{\partial q_\beta}\right)\dot{q}_\beta\right)\dot{q}_\gamma\right]$$

$$-\frac{1}{2}\frac{\partial}{\partial \dot{q}_\kappa}\left[\sum_{\gamma=1}^{\text{gen}}\left(\sum_{\beta=1}^{\text{gen}}\left(\sum_{i=1}^{n}\frac{\partial H_{\beta\gamma i}}{\partial q_\alpha} + \sum_{j=1}^{m}\int_0^{l_j}\frac{\partial \hat{H}_{\beta\gamma j}}{\partial q_\alpha}\,dy + \frac{\partial \tilde{H}_{\beta\gamma}}{\partial q_\alpha}\right)\dot{q}_\beta\right)\dot{q}_\gamma\right]$$

$$= \left[\sum_{\gamma=1}^{\text{gen}}\left(\sum_{i=1}^{n}\frac{\partial H_{\alpha\gamma i}}{\partial q_\kappa} + \sum_{j=1}^{m}\int_0^{l_j}\frac{\partial \hat{H}_{\alpha\gamma j}}{\partial q_\kappa}\,dy + \frac{\partial \tilde{H}_{\alpha\gamma}}{\partial q_\kappa}\right)\dot{q}_\gamma + \sum_{\beta=1}^{\text{gen}}\left(\sum_{i=1}^{n}\frac{\partial H_{\alpha\kappa i}}{\partial q_\beta} + \sum_{j=1}^{m}\int_0^{l_j}\frac{\partial \hat{H}_{\alpha\kappa j}}{\partial q_\beta}\,dy + \frac{\partial \tilde{H}_{\alpha\kappa}}{\partial q_\beta}\right)\dot{q}_\beta\right]$$

$$-\frac{1}{2}\left[\sum_{\gamma-1}^{\text{gen}}\left(\sum_{i=1}^{n}\frac{\partial H_{\kappa\gamma i}}{\partial q_\alpha} + \sum_{j-1}^{m}\int_0^{l_j}\frac{\partial \hat{H}_{\kappa\gamma j}}{\partial q_\alpha}\,dy + \frac{\partial \tilde{H}_{\kappa\gamma}}{\partial q_\alpha}\right)\dot{q}_\gamma + \sum_{\beta=1}^{\text{gen}}\left(\sum_{i=1}^{n}\frac{\partial H_{\beta\kappa i}}{\partial q_\alpha} + \sum_{j=1}^{m}\int_0^{l_j}\frac{\partial \hat{H}_{\beta\kappa j}}{\partial q_\alpha}\,dy + \frac{\partial \tilde{H}_{\beta\kappa}}{\partial q_\alpha}\right)\dot{q}_\beta\right]$$

$$= \sum_{\gamma=1}^{\text{gen}}\left(\sum_{i=1}^{n}\left(\frac{\partial H_{\alpha\gamma i}}{\partial q_\kappa} + \frac{\partial H_{\alpha\kappa i}}{\partial q_\gamma}\right) + \sum_{j=1}^{m}\int_0^{l_j}\left(\frac{\partial \hat{H}_{\alpha\gamma j}}{\partial q_\kappa} + \frac{\partial \hat{H}_{\alpha\kappa j}}{\partial q_\gamma}\right)dy + \left(\frac{\partial \tilde{H}_{\alpha\gamma}}{\partial q_\kappa} + \frac{\partial \tilde{H}_{\alpha\kappa}}{\partial q_\gamma}\right)\right)\dot{q}_\gamma - \frac{1}{2}\cdot 2\cdot\sum_{\gamma=1}^{\text{gen}}$$

$$\times\left(\sum_{i=1}^{n}\frac{\partial H_{\kappa\gamma i}}{\partial q_\alpha} + \sum_{j=1}^{m}\int_0^{l_j}\frac{\partial \hat{H}_{\kappa\gamma j}}{\partial q_\alpha}\,dy + \frac{\partial \tilde{H}_{\kappa\gamma}}{\partial q_\alpha}\right)\dot{q}_\gamma$$

$$= 2D_{\alpha\kappa}; \qquad \alpha, \kappa = 1, \cdots \text{gen.}$$

(iii) Proof of the skew-symmetric property: By applying the chain rule:

$$\dot{M}_{\alpha\kappa} = \sum_{\gamma=1}^{\text{gen}}\frac{\partial M_{\alpha\kappa}}{\partial q_\gamma}\dot{q}_\gamma = \sum_{\gamma=1}^{\text{gen}}\left(\sum_{i=1}^{n}\frac{\partial H_{\alpha\kappa i}}{\partial q_\gamma} + \sum_{j=1}^{m}\int_0^{l_j}\frac{\partial \hat{H}_{\alpha\kappa j}}{\partial q_\gamma}\,dy + \frac{\partial \tilde{H}_{\alpha\kappa}}{\partial q_\gamma}\right)\dot{q}_\gamma$$

$$\therefore W_{\alpha\kappa} = \dot{M}_{\alpha\kappa} - 2\frac{1\partial Q_\alpha}{2\partial \dot{q}_\kappa} = \dot{M}_{\alpha\kappa} - 2D_{\alpha\kappa}$$

$$= \sum_{\gamma=1}^{\text{gen}}\left(\sum_{i=1}^{n}\left(\frac{\partial H_{\alpha\kappa i}}{\partial q_\gamma} - \frac{\partial H_{\alpha\kappa i}}{\partial q_\gamma} - \frac{\partial H_{\alpha\gamma i}}{\partial q_\kappa} + \frac{\partial H_{\kappa\gamma i}}{\partial q_\alpha}\right) + \sum_{j=1}^{m}\int_0^{l_j}\left(\frac{\partial \hat{H}_{\alpha\kappa j}}{\partial q_\gamma} - \frac{\partial \hat{H}_{\alpha\kappa j}}{\partial q_\gamma} - \frac{\partial \hat{H}_{\alpha\gamma j}}{\partial q_\kappa} + \frac{\partial \hat{H}_{\kappa\gamma j}}{\partial q_\alpha}\right)dy + \left(\frac{\partial \tilde{H}_{\alpha\kappa}}{\partial q_\gamma} - \frac{\partial \tilde{H}_{\alpha\kappa}}{\partial q_\gamma} - \frac{\partial \tilde{H}_{\alpha\gamma}}{\partial q_\kappa} + \frac{\partial \tilde{H}_{\kappa\gamma}}{\partial q_\alpha}\right)\right)\dot{q}_\gamma$$

$$= \sum_{\gamma=1}^{\text{gen}} \left(\sum_{i=1}^{n} \left(\frac{\partial H_{\kappa\gamma i}}{\partial q_\alpha} - \frac{\partial H_{\alpha\gamma i}}{\partial q_\kappa} \right) + \sum_{j=1}^{m} \int_0^{l_j} \times \left(\frac{\partial \hat{H}_{\kappa\gamma j}}{\partial q_\alpha} - \frac{\partial \hat{H}_{\alpha\gamma j}}{\partial q_\kappa} \right) dy + \left(\frac{\partial \tilde{H}_{\kappa\gamma}}{\partial q_\alpha} - \frac{\partial \tilde{H}_{\alpha\gamma}}{\partial q_\kappa} \right) \right) \dot{q}_\gamma.$$

Because $H_{\alpha\kappa i}$, $\hat{H}_{\alpha\kappa j}$, $\tilde{H}_{\alpha\kappa}$ are all symmetric matrices, interchange α and κ. $\therefore$ $W_{\alpha\kappa} = -W_{\kappa\alpha}$ are skew-symmetric. QED.

3. Structure of a Symbolic Program for Dynamics Computation

Theorem 1 gives the dynamics of arbitrary rigid/flexible-link planar serial robot arms in a very convenient form for symbolic computation. Any mode shape function can be specified. Computer-automated symbolic expansion of the dynamic equations can be accomplished using a program written in Mathematica, an interactive software system that incorporates a high-level programming language. It uses symbolic expressions to provide a very general representation of mathematical and other structures. The software performs three basic types of computations: numerical, symbolic, and graphic. The implementation steps of our program follow the equations in Section 2. By inputting some specific data, the symbolic program can automatically derive the dynamic equations for a given planar flexible-link arm. The required user input is as follows:

1. Numbers of rigid links and flexible links.
2. The sequential order of the rigid links and flexible links. For example, number 1 and 3 are rigid links, and 2 and 4 are flexible links, The format looks like rig $= \{1,3\}$, and flex $= \{2,4\}$. If the system is comprised of all rigid links, then flex $= \{0\}$.
3. Number of flexible modes per link.
4. Mode shape function desired for each flexible link (e.g., pinned-pinned, clamped-free).

This program can be implemented in a PC, Next, Workstation, or mainframe environment. However, running the symbolic calculations is slow and requires large memory space. The CPU time required to find the solution is highly dependent on the integration process, especially when trigonometric and hyperbolic functions appear in the system. Therefore, the clamped mode assumption takes much longer CPU time than the pinned-pinned mode assumption, as it contains more trigonometric and hyperbolic functions. Fortunately, in the UNIX operating system, the program can run in the background. The result of the program will probably be too complicated to understand in its expanded form, but the "Simplify" function in Mathematica can reduce the matrix elements to their simplest form.

The symbolic computer-generated dynamic equations program concept is described as a system of several groups and a constructed database. Each group consists of values in a table list. During execution, the program checks the table list on each calculation step and retrieves the required information from the database. Our program has the following advantages:

1. This is an interactive symbolic program, so it is very general and flexible for the user.
2. The symbolic computation can relieve the user from the tedious tasks of analytic calculations and can increase the reliability of the derived equations.
3. Changing the number of rigid links and flexible links, or the mode shape functions, is very simple.
4. The method uses well-known theoretical procedures (Section 2) that are similar to those used in the rigid link approach.

4. Case Studies

4.1. *Rigid Two-Link Arm Example*

The rigid link model can be considered a special case in this symbolic program. Just set the "number of flexible links" and their corresponding variables equal to 0. Then, the rigid link manipulator dynamic equations are provided in the output of the program. To verify the correctness of the equations generated in this symbolic program, the well-known two-link revolute joint arm is compared to Spong and Vidyasagar (1989). The resulting equations are compared term by term and found to be exactly the same. Morever, our program generated $D(q,\dot{q})$ so that $\dot{M} - 2D$ is a skew-symmetric matrix.

4.2. *Flexible One-Link Arm Example*

The program can be used to examine a purely flexible link robot simply by setting the "numbers of rigid links" equal to 0. One flexible link, first with a pinned-pinned mode and then a clamped-free shape function, was considered. Four different flexible mode numbers (2, 5, 10, and 15) were computed for the link deflection. Development of the planar case for a simple revolute joint (rotating about the x-axis) and one flexible link follows:

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta_1 & -S\theta_1 \\ 0 & S\theta_1 & C\theta_1 \end{bmatrix}$$

$$P = Rh = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta_1 & -S\theta_1 \\ 0 & S\theta_1 & C\theta_1 \end{bmatrix} \begin{bmatrix} 0 \\ y \\ \Delta z \end{bmatrix},$$

$$\Delta z = \sum_{k=1}^{s} \phi_{kz}(y) V_k(t)$$

$$\dot{P} = \begin{bmatrix} \mathbf{0} \\ -y\dot{\theta}_1 S\theta_1 - \Delta\dot{z}S\theta_1 - \Delta z\dot{\theta}_1 C\theta_1 \\ y\dot{\theta}_1 C\theta_1 + \Delta\dot{z}C\theta_1 - \Delta z\dot{\theta}_1 S\theta_1 \end{bmatrix}$$

$$J = \begin{bmatrix} 0 & 0 \cdots 0 \\ -yS\theta_1 - \Delta zC\theta_1 & -(\phi_{1z} \cdots \phi_{sz})S\theta_1 \\ yC\theta_1 - \Delta zS\theta_1 & (\phi_{1z} \cdots \phi_{sz})C\theta_1 \end{bmatrix},$$

$$M = \int_0^{l_1} \rho A J^T J \, dy = \begin{bmatrix} M_{RR} & M_{RF} \\ M_{RF}^T & M_{FF} \end{bmatrix}$$

$$M_{RR} = \rho A \int_0^{l_1} \left(\left(yS\theta_1 + \sum_{k=1}^{s} \phi_{kz} V_k C\theta_1 \right)^2 + \left(yC\theta_1 - \sum_{k=1}^{s} \phi_{kz} V_k S\theta_1 \right)^2 \right) dy$$

$$M_{RF} = \rho A \int_0^{l_1} [y\phi_{1z} \cdots y\phi_{sz}] \, dy$$

where

$$M_{FF} = \rho A \int_0^{l_1} \begin{bmatrix} \phi_{1z}^2 \cdots & \cdots \phi_{1z}\phi_{sz} \\ \cdot & \cdot \\ \cdot & \cdot \\ \phi_{sz}\phi 1z \cdots & \cdots \phi_{sz}\phi_{sz} \end{bmatrix} dy$$

In the above derivation procedure, the effects of the flexible mode have been included in the M_{RR} term. This is contrary to the result of Kwon and Book (1990), which did not account for this mode effect and neglected the full nonlinear interactions between rigid and flexible component of arm dynamics. Moreover, M_{FF} is a positive definite symmetric constant matrix, and

$$M_{FF}(15 \text{ modes}) = \begin{bmatrix} M_{FF}(10 \text{ modes}) & \cdot \\ \cdot & \cdot \end{bmatrix}.$$

That is to say, the lower modes will be a partial block in the inertia matrix using higher modes. Similarly, the stiffness matrix (U) will have this same property for any different mode shape functions.

From the dynamic view point, the eigenvalues of $M_{FF}^{-1}U_{FF}$ are proportional to the resonant frequency (ω_s) of the flexible modes, and the amplitude is of the form $() \cdot e^{-\omega_s t}$ (Smith and Whaley 1989). Using our symbolic software (link properties: EI = 4120 lbf·in^2, L = 47 inches, ρA = 2.54e−4 * 0.14 lbf·s^2/in^2, tip mass = 0.1 lbm), the eigenvalues of $M_{FF}^{-1}U_{FF}$ for 15 flexible modes in pinned-pinned mode shape functions are: 2320.64, 1.87e5, 3.70e5, 5.92e5, 1.44e6, 2.99e6, 5.55e6, 9.47e6, 1.51e7, 2.31e7, 3.38e7, 4.79e7, 6.60e7, 8.88e7, and 1.17e8. These results predict that the higher mode frequencies will be limited to a very large value, which means that the amplitude of the higher mode vibrations will tend to zero. This is consistent with the well-known theory that only the lower modes (especially the first mode) dominate vibration in the elastic beam.

If the selected output is the joint angle (rigid mode) and the output is constrained to zero (which implies that all of its time derivatives are zero), the zero dynamics are defined (Isidori 1989; Slotine and Li 1991). In the pinned-pinned mode, the zero dynamics (neglecting structural damping and gravity effects in the system) are nonminimum phase, since they contain stable and unstable values ($-58.89 \pm j76.06, 58.89 \pm j76.06$). In the clamped-free mode, the zero dynamics are marginally stable, since the zeros exist as conjugate pairs on the imaginary axis ($\pm j76.98, \pm j1.0914$). These effects are all clearly apparent in our results.

4.3. *One Rigid Link and One Flexible Link (RF) Example*

This example shows that the algorithm can be applied when the arm is planar, except for a last link that can move out of plane. Actually, this is a general 3D arm in this two-link case. When we set lrig = {1} and lflex = {2}, the two-link revolute robot arm with flexible dynamics in the second link can be modeled (i.e., tank turret gun barrel problem [Lewis et al. 1991]). The azimuth angle is θ_1, and the elevation angle is θ_2. It is assumed that the second link can flex in both the horizontal and vertical directions. The dynamic equations of the two modes, using first pinned-pinned mode shape functions and then clamped-free shape functions, are derived. The symbolic notation of the output is as follows: $u[r, s]$ means x-direction flexible mode coordinates, and $v[r, s]$ means z-direction flexible mode coordinates, r means link number, s means mode number, and mpd means payload mass. Thus, the dynamic model for the two-link robot with two assumed modes is presented, and the vector of generalized coordinates reduces to

$$q = \{\theta_1, \theta_2, u[2,1], u[2,2], v[2,1], v[2,2]\}^T.$$

The result shows that the flexible motion significantly influences the centrifugal and Coriolis terms, but has little effect on the inertia matrix. The conclusion is similar to that in Lee (1992). The final dynamic equation form is explicitly separated into inertia, centrifugal, and Coriolis, gravitational, and structural stiffness terms. The output format is more suitable for computer simulation and easier to read than the form shown in Low (1987), Low and

Vidyasagar (1988), and Li and Sankar (1992). The result becomes more generalized than that in De Luca and Siciliano (1991) and Wang and Vidyasagar (1992), which constrain the link to flex horizontally.

The zero dynamics are defined when the output (azimuth angle θ_1 and elevation angle θ_2) are constrained to zero. The zero dynamics are nonminimum phase when the pinned-pinned mode shape function is applied in the flexible link. The zero dynamics are marginally stable when clamped-free mode shape functions are applied in the flexible link. The situation is similar to the one-link case.

4.4. *Rigid-Flexible-Rigid Link (RFR) Example*

When we set lrig $= \{1, 3\}$ and lflex $= \{2\}$, the three-link revolute robot arm with a flexible link in the second link is modeled. It is very easy to adjust the sequential order of the link number. This is a serial three-link and three-revolute-joint angle problem. In the system, rigid-flexible links are similar to case 1 in Section 2, where the equivalent joint angle θ_2^* does not need to be modified. Simultaneously, the flexible-rigid links are the same as case 2, so the equivalent joint angle θ_3^* must be modified. Even though θ_3^* is a highly nonlinear term and becomes more complex and tedious in the differentiation and integration processes, the symbolic program still works successfully. In such case, an intermediate flexible link is shown on the robot system, and therefore, pinned-pinned or clamped-free mode shape function are no longer appropriate for the flexible link. To achieve a more exact model, we introduced the mass boundary conditions in the partial differential equation. The partial differential equation and boundary conditions for the clamped-loaded Euler-Bernoulli beam can be expressed as

$$EI\Delta z''''(y,t) + \rho A\ddot{\Delta z}(y,t) = 0, \tag{27}$$

$$\Delta z(0,t) = 0, \tag{28}$$

$$\Delta z'(0,t) = 0, \tag{29}$$

$$\Delta z''(0,t) = 0, \tag{30}$$

$$EI\Delta z'''(L,t) = m_s\ddot{\Delta z}(L,t), \tag{31}$$

where Δz is the deformation for the flexible link on the z-direction and defined as in (1), with y representing displacement along neutral axis of the link; m_s is the mass for the second rigid link plus payload mass.

The general solution form for the mode shape function in (27) has been found as Smith and Whaley (1989)

$$\phi_i(y) = c_1 \cos\beta_i y + c_2 \cosh\beta_i y + c_3 \sin\beta_i y + c_4 \sinh\beta_i y. \tag{32}$$

From the boundary conditions (28)–(31), we can solve c_1, c_2, c_3, c_4, and β_1. There are an infinite number of solutions for the variables β_1, but only those values corresponding to the dominant modes are of interest; i.e., if the beam has r dominant modes, then the r smallest positive $\beta_i s$ are considered. The dynamic equations using clamped-loaded mode shape functions (time-varying boundary conditions) are developed and shown in the Appendix. The Appendix shows that the dynamic equation is very complex and highly nonlinear. The suggestion for the user is to run the program in the "Background" mode and use the "Simplify" mode to simplify the matrix element by element. Some insignificant terms can be neglected manually.

5. Conclusion

An efficient algorithm for the computation of planar serial link robot manipulator dynamic equations with arbitrarily assigned rigid or flexible links has been developed. The equations were derived using Lagrange's formulation and elastic deflection with the assumed-mode method. Any mode shape function can be used. A direct method is presented to determine the Coriolis/centrifugal matrix D, such that $\dot{M} - 2D$ is a skew-symmetric matrix, with M the inertia matrix. The algorithm was implemented symbolically using a program written in Mathematica. Four examples were given to verify the correctness of the computer-generated dynamic equations. In all examples, the zero dynamics using the pinned-pinned model are unstable, and using the clamped-free model is marginally stable (since the eigenvalues are located on the imaginary axis). A prototype tank turret barrel manipulator is presently being built by the Advanced Controls and Sensors Group at the Automation and Robotics Research Institute (ARRI) to verify the system characteristics. The proposed work to verify the dynamic equations will contain real-time control using a DSP (Digital Signal Processing) board and LabView software, and using optical encoders and strain gauges to measure the joint angles and elastic deflection.

Recommendations for future work include reducing the computing time in the symbolic computation and determining the neglected infinite flexible mode model effects in the control system (e.g., reducing observation spillover from the neglected modes by filtering or making more measurements than necessary).

Appendix

Mass Matrix for Rigid-Flexible-Rigid Link (Clamped-Loaded Mode Function)

In[2] : = $M[[1, 1]]$

Out[2] = $Ix[1] + Ix[3] + mpd\,L[1]^2$

$$+ 2\,mpd\cos[th[2][t]]L[1]\,L[2] +$$
$$mpd\,L[2]^2 + 2\,mpd\cos\left[th[2][t] + th[3][t] + \frac{0.5497897482078318v[2,1][t]}{L[2]}\right]L[1]\,L[3] +$$
$$2\,mpd\cos\left[th[3][t] + \frac{0.5497897482078318v[2,1][t]}{L[2]}\right]L[2]\,L[3] +$$
$$mpd\,L[3]^2 + Lc[1]^2ms[1] + L[1]^2ms[3] +$$
$$2\cos[th[2][t]]L[1]\,L[2]ms[3] + L[2]^2ms[3] +$$
$$2\cos\left[th[2][t] + th[3][t] + \frac{0.5497897482078318v[2,1][t]}{L[2]}\right]L[1]\,Lc[3]$$
$$ms[3] + 2\cos\left[th[3][t] + \frac{0.5497897482078318v[2,1][t]}{L[2]}\right]L[2]\,Lc[3]$$
$$ms[3] + Lc[3]^2ms[3]$$
$$+A[2]L[1]^2L[2]\,ro[2] +$$
$$A[2]\cos[th[2][t]]L[1]\,L[2]^2ro[2] +$$
$$\frac{A[2]L[2]^3ro[2]}{3} -$$
$$1.09958mpd\,L[1]\sin[th[2][t]]\,v[2,1][t] -$$
$$1.09958\,L[1]\,ms[3]\sin[th[2][t]]\,v[2,1][t] -$$
$$0.413302\,A[2]\,L[1]\,L[2]\,ro[2]$$
$$\sin[th[2][t]]\,v[2,1][t] +$$
$$1.099579496415663\,mpd\,L[3]\sin\left[th[3][t] + \frac{0.5497897482078318v[2,1][t]}{L[2]}\right]$$
$$v[2,1][t] + 1.099579496415663\,Lc[3]\,ms[3]$$
$$\sin\left[th[3][t]\frac{0.5497897482078318v[2,1][t]}{L[2]}\right] +$$
$$0.3022687672344311\,mpd\,v[2,1][t]^2 +$$
$$0.3022687672344311\,ms[3]\,v[2,1][t]^2 +$$
$$0.07147226535009999A[2]L[2]ro[2]v[2,1[t]^2$$

$$\text{In[3]} := M[[1,2]]$$
$$\text{Out[3]} = mpd\cos[th[2][t]]\,L[1]\,L[2] + mpd\,L[2]^2 +$$
$$mpd\cos\left[th[2][t] + th[3][t] + \frac{0.5497897482078318\,v[2,1][t]}{L[2]}\right]L[1]\,L[3] +$$
$$2\,mpd\cos\left[th[3][t] + \frac{0.5497897482078318\,v[2,1][t]}{L[2]}\right]L[2]\,L[3] +$$
$$mpd\,L[3]^2 + \cos[th[2][t]]\,L[1]\,L[2]\,ms[3]$$
$$+ L[2]^2ms[3] +$$
$$\cos\left[th[2][t] + th[3][t] + \frac{0.5497897482078318\,v[2,1][t]}{L[2]}\right]L[1]\,L[3]$$
$$ms[3] + 2\cos\left[th[3][t] + \frac{0.5497897482078318\,v[2,1][t]}{L[2]}\right]L[2]\,Lc[3]$$
$$ms[3] + Lc[3]^2ms[3]$$
$$+ \frac{A[2]\cos[th[2][t]]\,L[1]\,L[2]^2ro[2]}{L[2]} +$$
$$\frac{A[2]\,L[2]^3ro[2]}{3}$$
$$- 0.54979\,mpd\,L[1]\sin[th[2][t]]\,v[2,1][t] -$$
$$0.54979\,L[1]\,ms[3]\sin[th[2][t]]\,v[2,1][t] +$$
$$0.206651\,A[2]\,L[1]\,L[2]\,ro[2]\sin[th[2][t]]\,v[2,1][t] +$$
$$1.099579496415663\,mpd\,L[3]\sin\left[th[3][t] + \frac{0.5497897482078318v[2,1][t]}{L[2]}\right]$$
$$v[2,1][t] + 1.099579496415663\,Lc[3]\,ms[3]$$
$$\sin\left[th[3][t]\frac{0.5497897482078318v[2,1][t]}{L[2]}\right]$$
$$v[2,1][t] +$$
$$0.3022687672344311\,mpd\,v[2,1][t]^2 +$$
$$0.3022687672344311\,ms[3]\,v[2,1][t]^2 +$$
$$0.07147226535009999\,A[2]L[2]ro[2]v[2,1[t]^2$$

$$\text{In[4]} := M[[1,3]]$$
$$\text{Out[4]} = Ix[3] + mpd\left(\cos\left[th[2][t] + th[3][t] + \frac{0.5497897482078318\,v[2,1][t]}{L[2]}\right]\right.$$
$$L[1]\,L[3] + \cos\left[th[3][t] + \frac{0.5497897482078318\,v[2,1][t]}{L[2]}\right]L[2]$$
$$L[3] + L[3]^2 + 0.5497897482078318\,L[3]$$
$$\sin\left[th[3][t]\right.$$

$$+\frac{0.5497897482078318\,v[2,1][t]}{L[2]}\Big]v[2,1][t]\Big)+$$
$$ms[3]\Big(\cos\Big[th[2][t]+th[3][t]$$
$$+\frac{0.5497897482078318\,v[2,1][t]}{L[2]}\Big]L[1]$$
$$Lc[3]+\cos\Big[th[3][t]$$
$$+\frac{0.5497897482078318\,v[2,1][t]}{L[2]}\Big]L[2]\,Lc[3]+$$
$$Lc[3]^2+0.5497897482078318\,Lc[3]$$
$$\sin\Big[th[3][t]$$
$$+\frac{0.5497897482078318\,v[2,1][t]}{L[2]}\Big]v[2,1][t]\Big)$$

$$\text{In[5]} := M[[1,4]]$$
$$\text{Out[5]} = \Big(0.2748948741039159$$
$$\times\Big(2.\,mpd\cos[th[2][t]]\,L[1]\,L[2]^2+$$
$$2.\,mpd\cos\Big[th[2][t]+th[3][t]$$
$$+\frac{0.5497897482078318\,v[2,1][t]}{L[2]}\Big]L[1]$$
$$L[3]+4.\,mpd\cos\Big[th[3][t]$$
$$+\frac{0.5497897482078318\,v[2,1][t]}{L[2]}\Big]L[2]$$
$$L[3]+2.\,mpd\,L[3]^2$$
$$+2.\cos[th[2][t]]\,L[1]\,L[2]\,ms[3]+$$
$$2.\,L[2]^2ms[3]+2.\cos\Big[th[2][t]+th[3][t]+$$
$$\frac{0.5497897482078318\,v[2,1][t]}{L[2]}\Big]L[1]\,Lc[3]\,ms[3]+$$
$$4.\cos\Big[th[3][t]$$
$$+\frac{0.5497897482078318\,v[2,1][t]}{L[2]}\Big]L[2]\,Lc[3]\,ms[3]+$$
$$2.\,Lc[3]^2ms[3]$$
$$+0.7517462583877994\,A[2]\cos[th[2][t]]\,L[1]\,L[2]^2$$
$$ro[2]+0.5509971691837638\,A[2]\,L[2]^3ro[2]+$$
$$1.099579496415663\,mpd\,L[3]$$
$$\sin\Big[th[3][t]$$
$$+\frac{0.5497897482078318\,v[2,1][t]}{L[2]}\Big]v[2,1][t]+$$
$$1.099579496415663\,Lc[3]\,ms[3]$$
$$\sin\Big[th[3][t]$$
$$+\frac{0.5497897482078318\,v[2,1][t]}{L[2]}\Big]v[2,1][t]\Big)\Big)/L[2]$$

$$\text{In[6]} := M[[2,2]]$$
$$\text{Out[6]} = mpd\,L[2]^2+2\,mpd\cos\Big[th[3][t]$$
$$+\frac{0.5497897482078318\,v[2,1][t]}{L[2]}\Big]L[2]$$
$$L[3]+mpd\,L[3]^2+L[2]^2ms[3]+$$
$$2\cos\Big[th[3][t]$$
$$+\frac{0.5497897482078318\,v[2,1][t]}{L[2]}\Big]L[2]\,Lc[3]\,ms[3]+$$
$$Lc[3]^2ms[3]+\frac{A[2]\,L[2]^3ro[2]}{3}+$$
$$1.099579496415663\,mpd\,L[3]\sin\Big[th[3][t]$$
$$+\frac{0.5497897482078318\,v[2,1][t]}{L[2]}\Big]$$
$$v[2,1][t]+1.099579496415663\,Lc[3]\,ms[3]$$
$$\sin\Big[th[3][t]$$
$$+\frac{0.5497897482078318\,v[2,1][t]}{L[2]}\Big]v[2,1][t]+$$
$$0.3022687672344311\,mpd\,v[2,1][t]^2+$$
$$0.3022687672344311\,ms[3]\,v[2,1][t]^2+$$
$$0.07147226535009999\,A[2]\,L[2]\,ro[2]\,v[2,1][t]^2$$

$$\text{In[7]} := M[[2,3]]$$
$$\text{Out[7]} = mpd\Big(\cos\Big[th[3][t]$$
$$+\frac{0.5497897482078318\,v[2,1][t]}{L[2]}\Big]L[2]\,L[3]+$$
$$L[3]^2+0.5497897482078318\,L[3]$$
$$\sin\Big[th[3][t]$$
$$+\frac{0.5497897482078318\,v[2,1][t]}{L[2]}\Big]v[2,1][t]\Big)+$$
$$ms[3]\Big(\cos\Big[th[3][t]$$
$$+\frac{0.5497897482078318\,v[2,1][t]}{L[2]}\Big]L[2]\,Lc[3]+$$

$$Lc[3]^2 + 0.5497897482078318\, Lc[3]$$
$$\sin\left[th[3][t] + \frac{0.5497897482078318\, v[2,1][t]}{L[2]}\right] v[2,1][t]\Bigg)$$

$$\text{In}[8] := M[[2,4]]$$
$$\text{Out}[8] = \Bigg(0.5497897482078318 \times \Bigg(1.\, mpd\, L[2]^2 + 2.\, mpd \cos\left[th[3][t] + \frac{0.5497897482078318\, v[2,1][t]}{L[2]}\right] L[2]\, L[3] + 1.\, mpd\, L[3]^2 + 1.\, L[2]^2 ms[3] + 2.\cos\left[th[3][t] + \frac{0.5497897482078318\, v[2,1][t]}{L[2]}\right] \times L[2]\, Lc[3]\, ms[3] + 1.\, Lc[3]^2 ms[3] + 0.2754985845918819\, A[2]\, L[2]^3\, ro[2] + 0.5497897482078318\, mpd\, L[3] \sin\left[th[3][t] + \frac{0.5497897482078318\, v[2,1][t]}{L[2]}\right] v[2,1][t] + 0.5497897482078318\, Lc[3]\, ms[3] \sin\left[th[3][t] + \frac{0.5497897482078318\, v[2,1][t]}{L[2]}\right] v[2,1][t]\Bigg)\Bigg) / L[2]$$

$$\text{In}[9] := M[[3,3]]$$
$$\text{Out}[9] = Ix[3] + mpd\, L[3]^2 + Lc[3]^2 ms[3]$$

$$\text{In}[10] := M[[3,4]]$$
$$\text{Out}[10] = 0.5497897482078318\, mpd \cos \times \left[th[3][t] \frac{0.5497897482078318\, v[2,1][t]}{L[2]}\right] L[3] + \frac{0.5497897482078318\, mpd\, L[3]^2}{L[2]}\Bigg] + 0.5497897482078318 \cos\left[th[3][t] + \frac{0.5497897482078318\, v[2,1][t]}{L[2]}\right] Lc[3]$$
$$ms[3] + \frac{0.5497897482078318\, Lc[3]^2 ms[3]}{L[2]}$$

$$\text{In}[11] := M[[4,4]]$$
$$\text{Out}[11] = 0.3022687672344311\, mpd + \Bigg(0.6045375344688622\, mpd \cos\left[th[3][t] + \frac{0.5497897482078318\, v[2,1][t]}{L[2]}\right] L[3]\Bigg) / L[2] + \frac{0.3022687672344311\, mpd\, L[3]^2}{L[2]^2} + 0.3022687672344311\, ms[3] + \Bigg(0.6045375344688622 \cos\left[th[3][t] + \frac{0.5497897482078318\, v[2,1][t]}{L[2]}\right] Lc[3] ms[3]\Bigg) / L[2] + \frac{0.3022687672344311\, L[3]^2 ms[3]}{L[2]^2} + 0.07147226535009999\, A[2]\, L[2]\, ro[2]$$

Acknowledgments

The research described in this article was supported by Texas Advanced Technology Grant 003656-008 and National Science Foundation Grant MSS-9114009.

References

Cetinkunt, S., and Book, W. J. 1989. Symbolic modeling and dynamic simulation of robotic manipulators with compliant links and joints. *Robot. Computer-Integrated Manufacturing* 5(4):301–310.

Cetinkunt, S., and Ittop, B. 1992. Computer-automated symbolic modeling of dynamics of robotic manipulators with flexible links. *IEEE Trans. Robot. Automation* 8(1):94–105.

De Luca, A., and Siciliano, B. 1991. Closed-form dynamic model of planar multilink lightweight robots. *IEEE Trans. Sys. Man Cybernet.* 21(4):826–839.

Faessler, H. 1986. Computer-assisted generation of dynamical equations for multibody systems. *Int. J. Robot. Res.* 5(3):129–141.

Gorman, D. J. 1975. *Free Vibration Analysis of Beams and Shafts*. New York: John Wiley & Sons.

Hollerbach, J. M. 1980. A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity. *IEEE Trans. Sys. Man Cybernet.* SMC-10(11):730–736.

Isidori, A. 1989. *Nonlinear Control Systems*. New York: Springer-Verlag.

Kwon, D. S., and Book, W. J. 1990. An inverse dynamic method yielding flexible manipulator state trajectories. *Proc. of American Control Conf.*, San Diego, CA, pp. 186–193.

Lewis, F. L., Dawson, D. M., Lin, J., and Liu, K. 1991. Tank gun-pointing control with barrel flexibility effects. *ASME Publication Modelling and Control of Compliant and Rigid Motion Sys.* DSC-31:47–54.

Lee, S. H. 1992. Robust control of a flexible link robot and rigid link robot: Theory, simulation, and experiment. Ph.D. thesis, School of Mechanical Engineering, Georgia Institute of Technology.

Leu, M. C., and Hemati, N. 1986. Automated symbolic derivation of dynamic equations of motion for robotic manipulators. *ASME J. Dynam. Sys. Measurement Control* 108:172–179.

Li, C. J., and Sankar, T. S. 1992. A systematic method of dynamics for flexible robot manipulators. *J. Robot. Sys.* 9(7):861–891.

Lin, J. and Lewis, F. L. 1993. Improved measurement/estimation techniques for flexible link robot arm control. *Proc. of 32nd IEEE Conf. on Decision and Control.* San Antonio, TX, pp. 627–632.

Low, K. H. 1987. A systematic formulation of dynamic equations for robot manipulators with elastic links. *J. Robot. Sys.* 4(3):435–456.

Low, K. H., and Vidyasagar, M. 1988. A Lagrangian formulation of the dynamic model for flexible manipulator systems. *ASME J. Dynam. Sys. Measurement Control* 110:175–181.

Maeder, R. 1990. *Programming in Mathematica.* Redwood City: Addison-Wesley.

Slotine, J-J. E., and Li, W. 1991. *Applied Nonlinear Control.* Englewood Cliffs, NJ: Prentice-Hall.

Smith, J., and Whaley, W. 1989. *Vibration of Mechanical and Structural Systems.* New York: Harper & Row.

Spong, M. W. and Vidyasagar, M. 1989. *Robot Dynamics and Control.* New York: John Wiley & Sons.

Wang, D., and Vidyasagar, M. 1992. Modeling a class of multilink manipulators with the last link flexible. *IEEE Trans. Robot. Automation* 8(1):33–41.

Wolfram, S. 1991. *Mathematica: A System for Doing Mathematics by Computer.* Redwood City, CA: Addison-Wesley.

Yuan, B. S. 1989. Adaptive strategies for controls of flexible arms. Ph.D. thesis. School of Mechanical Engineering, Georgia Institute of Technology.