of the error between measured and reconstructed cross-sections is lower than 2%.

A potential use of this method is to develop applications in CAD/CAM. If no information is given on the location of the revolution axis, then using a zoom lens can be an interesting approach. But we can also think about an accurate modeling system with a fixed camera and a devoted place to put the object. In this case the revolution axis which remains fixed for all objects, is located during the calibration step, and modeling the object of revolution amounts to solve the inverse perspective problem of limb points projection.

REFERENCES

[1] C. Carmona, "Etude de la stéréovision axiale. Modélisation mathématique et résoluion algorithmique," Thèse de Doctorat de l'Institut National Polythechnique de Toulouse, Dec. 1991.

[2] M. Dhome, J. T. Lapreste, G. Rives, and M. Richetin, "Spatial localization of modelled objects of revolution in monocular perspective vision," in *Proc. ECCV '90*, 1990, pp. 475–485.

[3] R. Glachet, M. Dhome, and J. T. Lapreste, "Finding the perspective projection of an axis of revolution," *Patt. Recognit. Lett.*, vol. 12, no. 11, pp. 693–700, Nov. 1991.

[4] J. M. Lavest, R. Glachet, M. Dhome, and J. T. Lapreste, "Modelling solids of revolution by monocular vision," in *Proc. IEEE Comput. Vision and Patt. Recognit.*, 1991, pp 690–691.

[5] J. M. Lavest, "Stéréovision axiale par zoom pour la robotique," Thèse de Doctorat de l'Université Blaise-Pascal, no. 445, Clermont-Ferrand, France, July 1992.

[6] J. M. Lavest, G. Rives, and M. Dhome, "3D reconstruction by zooming," *IEEE Trans. Robot. and Automat.*, vol. 9, no. 2, pp. 196–207, Apr. 1993.

[7] J. Ponce, D. Chellberg, and W. Mann, "Invariant properties of straight homogeneous generalized cylinders and their contours," *IEEE Trans. Patt. and Mach. Intell.*, vol. 11, no. 9, pp. 951–966, Sept. 1989.

[8] J. Ponce and D. J. Kriegman, "On recognizing and positioning curved 3D objects from image contours," in *Proc. IEEE Workshop on Interpretation of 3D Scenes*, 1989, pp. 61–67.

[9] P. Saint-Marc and G. Medioni, "B-spline contour representation and symmetry detection," in *Proc. ECCV '90*, 1990, pp. 604–606.

[10] K. Tarabanis, R. Y. Tsai, and D. S. Goodman, "Modeling of a computer-controlled zoom lens," in *Proc. IEEE Int. Conf. Robot. and Automat.*, 1992, pp. 1545–1551.

[11] R. Vaillant, and O. D. Faugeras, "Using extremal boundaries for 3-D object modeling," in *IEEE Trans. Patt. Anal. and Mach. Intell.*, vol. 14, no. 2, pp. 157–173, Feb. 1992.

[12] A. G. Wiley and K. W. Wong, "Metric aspect of zoom vision," in *Proc. SPIE Symposium on Close Range Photogrammetry Meets Machine Vision*, 1990, vol. 1395, pp. 112-118.

Symbolic Construction of Models for Multibody Dynamics

Harry G. Kwatny and Gilmer L. Blankenship

***Abstract*—New algorithms are presented for deriving joint kinematic relations and these are integrated with Poincaré's form of Lagrange's equations to generate the dynamical equations of motion for rigid multibody chains. Software is described which performs all of the required symbolic constructions. Examples are given.**

I. INTRODUCTION

Computer assembly of simulation models has become recognized as an important engineering tool. However, relatively little consideration has been given to other applications of computer generated models such as: nonlinear control system design, construction of Lyapunov functions, bifurcation analysis, identification of symmetries and nonlinear system reduction. These require an explicit symbolic representation of systems not necessary for simulation purposes. The growing sophistication of symbolic programming languages such as Mathematica and Maple place such applications within reach for systems of current interest.

Recently, we described symbolic software for the design of nonlinear tracking and adaptive control laws [1]. We illustrated these methods with relatively simple multibody models of vehicle and robotic subsystems. In addition to control system design tools, the software included automatic generation of model equations via Lagrange's equations and generation of simulation code (in C or FORTRAN). In this paper, we provide new algorithms and describe software suitable for more complex multibody structures. Our model building process begins with primitive definitions of individual joints, bodies and other components and generates the equations of motion in terms of Poincaré's form of Lagrange's equations [2], [3], sometimes called Lagrange's equations for quasicoordinates [4].

The key to this process is the formulation of individual joint models including the selection of joint parameters and the assembly of the joint configuration matrix, the joint kinematic equation and the joint map matrix. In Section II we describe a geometric formulation of the joint representation problem and provide explicit constructions for both simple and compound joints. These are the main contributions of the paper. Once the individual joint models are obtained, a second important computation is the construction of the kinetic energy function or, equivalently, the system inertia matrix. Our formulation employs an algorithm proposed by Rodriguez and Jain [5]–[7]. In [8], we explain how that systematic construction is naturally integrated with the formulation of Poincaré's equations. We summarize the essential ideas behind the derivation of Poincaré's equations in Section III and describe our symbolic implementation of that process. In Section IV we give a nontrivial example of a six degree of freedom robot—a system of the same kinematic complexity as the Puma 560 described in [9] and the space shuttle remote manipulator system [10].

Manuscript received May 24, 1993; revised January 21, 1994. This work was supported in part by US Army TACOM under Contract no. DAAE07-C-R022.

H. G. Kwanty is with the Department of Mechanical Engineering & Mechanics, Drexel University, Philadelphia, PA 19104 USA; Email: harry.kwatny@coe.drexel.edu.

G. L. Blankenship is with the Department of Electrical & Computer Engineering, University of Maryland, College Park, MD 20742 USA; Email: gilmer@eng.umd.edu.

IEEE Log Number 9409075.

During the past decade a number of powerful software programs have become available (e.g. ADAMS, DYMAC, DADS, TREETOPS [11], [12]) which have the ability to assemble simulation models for reasonably complicated multibody systems. Consistent with their focus on simulation, these packages generate implicit models and consequently provide only limited support for analytical supplements to simulation or for control system design [12]. Computer derivation of the explicit equations of motion for multibody systems has been previously considered by other investigators including Leu and Hemati [13] and Cetinkunt and Ittoop [14]. Our approach extends that work in two important respects. First, we admit a more general class of joint models in which the joint parameterization and all relevant joint kinematic relations are derived directly from the specific joint definition—as opposed to prescribing them beforehand. Second, we use Poincaré's form of Lagrange's equations which allows maximum freedom of choice for velocity coordinates. That can contribute to substantially simplified dynamical equations.

II. Joint Kinematics

Efficient characterization of joint kinematics is a key element in modeling articulated multibody system dynamics. Bodies are linked together by joints which restrict relative motion between them. Multibody system configuration coordinates naturally include the joint configuration coordinates. The critical elements of joint modeling are the definition of joint parameters, the representation of joint configuration matrix in terms of them, and the subsequent derivation of two fundamental relationships. The first is a differential equation, typically called the *joint kinematic equation*, which defines the joint coordinate derivatives in terms of the joint quasivelocities. The second is a (joint) parameter-dependent map, called the *joint map*, which specifies the velocity (angular and linear) change across a joint in terms of the joint quasivelocities.

Composing the configuration matrix, the kinematic equation and the joint map involves tedious algebraic constructions for all but the simplest of joints. Real joints tend not to be nice, e.g., motion axes are not necessarily orthogonal, which complicates the calculations. In this paper we describe the computer assembly of these relations for a reasonably large class of joints.

First we summarize the basic geometric formalism for dealing with joints. Then we distinguish between "simple" and "compound" joints and focus initially on simple joints. We explicitly provide one parameterization for them. In appropriate special cases the joint parameters turn out to be Euler parameters so we may say that our parameterization is Euler-like. We also describe a construction of the kinematic equation which has been implemented in Mathematica. Examples are given which show the computer generated results for some standard simple joints. For simple joints the joint map is essentially the definition of the joint and need not be generated. This is not the case, however, for compound joints. The kinematic equation is constructed as for simple joints, but for compound joints the joint map is also a nontrivial calculation. We describe a method for assembly of the joint map which has also been implemented in Mathematica. An example is given.

A. The Geometry of Joints

A joint constrains the relative motion between two bodies. We designate two rigid bodies and reference frames fixed within them s (space) and b (body). The configuration space $\mathcal{G}$ of relative motion between two unconstrained rigid bodies is the Special Euclidean group SE(3) consisting of all rotations and translations of R^3. SE(3) is the semi-direct product of the rotation group SO(3) with the vector group R^3, [15]. An element in SE(3) may be represented by a matrix

$$X = \begin{bmatrix} L^T & R \\ 0 & 1 \end{bmatrix}, \quad L \in \mathrm{SO}(3), \; R \in R^3. \tag{1}$$

Two successive relative motions X_1 and X_2 combine to yield

$$X = X_2 X_1 = \begin{bmatrix} L_2^T & R_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} L_1^T & R_1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} L_2^T L_1^T & L_2^T R_1 + R_2 \\ 0 & 1 \end{bmatrix}. \tag{2a}$$

The inverse of X is

$$X^{-1} = \begin{bmatrix} L & -R \\ 0 & 1 \end{bmatrix}. \tag{2b}$$

In general geometric terms, a joint is characterized by a relation on the tangent bundle $T\mathcal{G}$. Such a relation is usually expressed in local coordinates by an equation of the type

$$f(q, \dot{q}) = 0 \tag{3a}$$

where $f: T\mathcal{G} \to R^k$. Natural constraints almost always occur on one of two forms:

$$f(q) = 0 \tag{3b}$$

in which only the coordinates appear, or

$$F(q)\dot{q} = 0 \tag{3c}$$

in which the coordinate velocities appear linearly. Equation (3b) defines a submanifold of $\mathcal{G}$ which identifies admissible configurations. Constraints of this from are called *geometric* constraints because they restrict the relative geometry of the two bodies. Constraints of the form (3c) are called *kinematic* because they restrict the relative velocity of two bodies. The geometric meaning of (3c) is highlighted by restating it as

$$\dot{q} \in \Delta(q) \tag{4}$$

where $\Delta(q)$ is a distribution on $\mathcal{G}$ defined as $\Delta(q) = \mathrm{Ker}[A(q)]$. If the constraint is of the form of (3c), then it is *holonomic* [16], [17] if the distribution $\Delta(q)$ is integrable. General conditions for integrability of a distribution are well known and given by the Frobenius theorem [15].

Since $T\mathcal{G}$ is isomorphic to $\mathcal{G} \times \mathbf{g}$, where $\mathbf{g}$ denotes the Lie algebra associated with $\mathcal{G}$, it is possible to characterize joint constraints which involve velocities (i.e., 3c) by a smooth map $f: \mathcal{G} \times \mathbf{g} \to R^k$ so that the joint is defined by equations of the form:

$$A(q)p = 0, \tag{5}$$

where $p \in \mathbf{g}$ and $A(q)$ is a linear operator on $\mathbf{g}$. The geometric meaning of (5) is

$$p \in \mathrm{Ker}[A(q)]. \tag{6}$$

Equation (5) is a more general and more convenient characterization of kinematic joints than (3c).

Let us take $\mathcal{G} = \mathrm{SE}(3)$ and consider the formal representation of objects belonging to its Lie algebra $\mathbf{g} = se(3)$. We can use either right or left translations on $\mathcal{G}$ to define $\mathbf{g}$. We choose left, so that

$$p := X^{-1}\dot{X} = \begin{bmatrix} L & -R \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{L}^T & \dot{R} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} L\dot{L}^T & L\dot{R} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \tilde{\omega}_b & v_b \\ 0 & 0 \end{bmatrix}. \tag{7}$$

TABLE I
EXAMPLES OF JOINT MAP MATRICES OF SIMPLE JOINTS

1 dof revolute joint body z–axis	2 dof universal joint body y,z–axis	1 dof prismatic joint body z–axis	1 dof screw joint body z–axis	2 dof cylindrical joint body z–axis
$\begin{bmatrix}0\\0\\1\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0&0\\1&0\\0&1\\0&0\\0&0\\0&0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\\0\\0\\1\end{bmatrix}$	$\begin{bmatrix}0\\0\\1\\0\\0\\s\end{bmatrix}$	$\begin{bmatrix}0&0\\0&0\\1&0\\0&0\\0&0\\0&1\end{bmatrix}$

Notice that in (7) we use the conventional notation, by which any vector $a \in R^3$ is converted into a skew-symmetric matrix $\tilde{a}(a)$:

$$\tilde{a}(a) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

Thus, we see that g is isomorphic to R^6 and we can consider an element p of g to be a pair of objects—body angular velocity and linear velocity—$(\tilde{\omega}_b, v_b)$ or, equivalently, (ω_b, v_b). When doing formal group calculations however, we use the matrix form shown in (7).

B. Simple Kinematic Joints

Kinematic joints are joints which are described by velocity constraints such as (3c) or (5). They are simple if the motion axes are fixed in (at least) one of the bodies—in which case the constraint can be formulated so that A is a constant (independent of the configuration). For lack of a general terminology we call such joints *simple kinematic joints*. We now focus on simple kinematic joints. It is convenient to define a matrix H whose columns form a basis for Ker$[A]$ so that

$$\mathrm{Ker}[A] = \mathrm{Im}[H], \quad H \text{ is of full rank } r = \dim \mathrm{Ker}[A]. \tag{8}$$

Solutions of (5) are of the form

$$p = H\beta, \quad \beta \in R^r \tag{9}$$

β represents the joint quasivelocity and r is the number of velocity degrees of freedom. H is called the *joint map matrix*.

The joint configuration is defined, in general, by the differential equations

$$\dot{X} = Xp \tag{10a}$$

or, equivalently

$$\dot{L} = -\tilde{\omega}_b L, \quad \dot{R} = L^T v_b. \tag{10b}$$

It is easy enough to replace ω_b and v_b by β using (9). Let H be partitioned so that H_1 contains the first 3 rows and H_2 the second three rows, then

$$\dot{X} = X\begin{bmatrix} \tilde{H}_1\beta & H_2\beta \\ 0 & 0 \end{bmatrix} \tag{11a}$$

or

$$\dot{L} = -(\tilde{H}_1\beta)L, \quad \dot{R} = L^T(H_2\beta). \tag{11b}$$

The joint kinematics are defined by (11). Given the quasivelocities β, (11) can be integrated to provide the relative translational position and rotation matrix of the two bodies. However, this representation may not be the most informative and it certainly provides more information than necessary since it locates the relative position in the six dimensional group SE(3) instead of the relevant subgroup. If the constraint is holonomic, precisely r dimensions would suffice. First, we provide a result for single degree of freedom joints.

Proposition 1: Consider a simple single degree of freedom joint with joint map matrix $H = h \in R^6$. Then the joint configuration matrix can be parameterized by a parameter $z \in R$ in the form:

$$\chi(z) = \begin{bmatrix} L^T(z) & R(z) \\ 0 & 1 \end{bmatrix} \tag{12a}$$

with

$$L(z) = e^{-\tilde{h}_1 z}, \quad R(z) = \int_0^z e^{\tilde{h}_1\sigma} h_2 d\sigma. \tag{12b}$$

Proof: Consider a general one degree of freedom joint in which H is composed of the single column h. Then the distribution $\Delta(X)$ on SE(3) consists of the single vector field

$$\begin{bmatrix} L^T\tilde{h}_1 & L^T h_2 \\ 0 & 0 \end{bmatrix}.$$

This is an integrable distribution and we seek the integral manifold which passes through the point

$$X_0 = \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix}.$$

The one dimensional manifold we seek can be characterized (at least locally) by a map $\chi: R \to$ SE(3). Let $z \in R$ be the parameter. Then we seek a solution to the differential equation

$$\frac{d\chi}{dz} = \begin{bmatrix} L^T\tilde{h}_1 & L^T h_2 \\ 0 & 0 \end{bmatrix}, \quad \chi(0) = X_0 \tag{13}$$

or equivalently

$$\frac{dL}{dz} = -\tilde{h}_1 L, \quad L(0) = I, \quad \text{and} \quad \frac{dR}{dz} = L^T h_2, \quad R(0) = 0 \tag{14}$$

so that (12a) and (12b) follow. □

We will give some explicit examples below.

Note that if H is composed of several columns, say r, then we can consider this joint as a sequence of r single column joints and compute $\chi_i(z_i)$ for each joint. Thus, we have

Corollary 1: Consider a simple joint with r degrees of freedom and joint map matrix $H = [h_1 \cdots h_r] \in R^{6\times r}$, then there is a parameter vector $z \in R^r$ and the joint configuration matrix can be expressed in the form

$$\chi(z) = \chi_r(z_r)\cdots\chi_2(z_2)\chi_1(z_1) \tag{15}$$

where each $\chi_i(z_i)$ is of the form of Proposition 1 with $h = h_i$.

We conclude that any simple kinematic joint is holonomic and, in fact, we have explicitly computed a local representation of its configuration manifold. Now, any motion results in a velocity $\dot{X} = Xp$. We wish to characterize this relation (locally) in terms of the rate of change of the joint parameters. In other words, we seek to relate $\dot{z}$ and β. The following proposition does that.

Proposition 2: Consider a simple joint with joint map matrix $H = [h_1 \cdots h_r] \in R^{6\times r}$, and suppose the joint is parameterized according to Proposition 1 and Corollary 1. Then the joint kinematic equation is

$$\dot{z} = \Gamma(z)\beta \tag{16}$$

where $\Gamma(z)$ is defined by the following algorithm:

1) For $j = 1, \ldots, r$ define $\mathcal{L}_j^T$ and $\mathcal{R}_j$

$$\mathcal{L}_j^T(z_j, \ldots, z_1) = L_j^T(z_j) + \mathcal{L}_{j-1}^T(z_{j-1}, \ldots, z_1), \mathcal{L}_0^T = I \tag{17a}$$

$$\mathcal{R}_j(z_j, \ldots, z_1) = L_j^T(z_j) + \mathcal{R}_{j-1}(z_{j-1}, \ldots, z_1), +R_j, \mathcal{R}_0 = I. \tag{17b}$$

2) Define $B(z)$

$$B(z) := \begin{bmatrix} b_{11} & \cdots & b_{1r} \\ b_{21} & \cdots & b_{2r} \end{bmatrix} \tag{18a}$$

$$\tilde{b}_{1i} := \mathcal{L}_{i-1}\tilde{h}_{i1}\mathcal{L}_{i-1}^T \tag{18b}$$

$$b_{2i} := \mathcal{L}_{i-1}\tilde{h}_{i1}\mathcal{R}_{i-1} + \mathcal{L}_{i-1}h_{i2}. \tag{18c}$$

3) Define $\Gamma(z)$

$$\Gamma(z) := [B^*(z)H], \quad B^*(z) \text{ denotes a left inverse of } B(z). \tag{19}$$

Proof: Any motion results in a velocity $\dot{X} = Xp$ which implies

$$\dot{X} = \Sigma \frac{\partial X}{\partial z_i}\dot{z}_i = X(z)p.$$

Now, we directly compute

$$\Sigma \frac{\partial X}{\partial z_i}\dot{z}_i = \Sigma X_r(z_r)\cdots X_{i+1}(z_{i+1}) \times \frac{dX_i}{dz_i} X_{i-1}(z_{i-1})\cdots X_1(z_1)\dot{z}_i$$

and premultiplying by X^{-1} we obtain

$$\Sigma[X_{i-1}(z_{i-1})\cdots X_1(z_1)]^{-1} X_i^{-1}(z_i) \times \frac{dX_i}{dz_i}[X_{i-1}(z_{i-1})\cdots X_1(z_1)]\dot{z}_i = p. \tag{20}$$

Notice that

$$X_i^{-1}(z_i)\frac{dX_i}{dz_i} = \begin{bmatrix} \tilde{h}_{i1} & h_{i2} \\ 0 & 0 \end{bmatrix}.$$

Also, define $U_j(z_j, \ldots, z_1), j = 1, \ldots, r$ by the recursion

$$U_j(z_j, \ldots, z_1) := X_j(z_j)U_{j-1}(z_{j-1}, \ldots, z_1), \quad \text{with} \quad U_1(z_1) = X_1(z_1) \tag{21}$$

so that (20) can be written

$$\Sigma U_{i-1}^{-1}\begin{bmatrix} \tilde{h}_{i1} & h_{i2} \\ 0 & 0 \end{bmatrix} U_{i-1}\dot{z}_i = p. \tag{22}$$

We can easily determine, from (18), that U_j is of the form

$$U_j = \begin{bmatrix} \mathcal{L}_j^T(z_j, \ldots, z_1) & \mathcal{R}_j(z_j, \ldots, z_1) \\ 0 & 1 \end{bmatrix}. \tag{23a}$$

with

$$\mathcal{L}_j^T(z_j, \ldots, z_1) = L_j^T(z_j)\mathcal{L}_{j-1}^T(z_{j-1}, \ldots, z_1), \mathcal{L}_0^T = I \tag{23b}$$

$$\mathcal{R}_j(z_j, \ldots, z_1) = L_j^T(z_j)\mathcal{R}_{j-1}(z_{j-1}, \ldots, z_1) + R_j, \mathcal{R}_0 = 0. \tag{23c}$$

Thus, (22) reduces to

$$\Sigma \begin{bmatrix} \mathcal{L}_{i-1}\tilde{h}_{i1}\mathcal{L}_{i-1}^T & \mathcal{L}_{i-1}\tilde{h}_{i1}\mathcal{R}_{i-1} + \mathcal{L}_{i-1}h_{i2} \\ 0 & 0 \end{bmatrix}\dot{z}_i = p = \begin{bmatrix} \tilde{H}_1\beta & H_2\beta \\ 0 & 0 \end{bmatrix}. \tag{24}$$

Each expression of the form $\mathcal{L}_{i-1}\tilde{h}_{i1}\mathcal{L}_{i-1}^T$ is an antisymmetric matrix so we can define $b_{1i} \in R^3$ such that

$$\tilde{b}_{1i} := \mathcal{L}_{i-1}\tilde{h}_{i1}\mathcal{L}_{i-1}^T.$$

We also define

$$b_{2i} := \mathcal{L}_{i-1}\tilde{h}_{i1}\mathcal{R}_{i-1} + \mathcal{L}_{i-1}h_{i2}.$$

Then (24) can be written

$$B(z)\dot{z} = H\beta, B(z) := \begin{bmatrix} b_{11} & \cdots & b_{1r} \\ b_{21} & \cdots & b_{2r} \end{bmatrix}.$$

Let B^* denote the left inverse of B—which exists on a neighborhood of $z = 0$ because $B(0) = H$ is of full rank. Then

$$\dot{z} = [B^*(z)H]\beta = \Gamma(z)\beta, \quad \Gamma(z) := [B^*(z)H]. \quad \square$$

C. Computer Implementation

There are two useful constructions for simple joints: (1) assembly of the configuration matrix, $X(z)$, and (2) assembly of the kinematic matrix $\Gamma(z)$. Correspondingly, we have two Mathematica functions to implement these calculations; `XXeuc` and `GammaKin`. The input to these functions is the $6 \times r$ joint map matrix, H, and a list of r names for the joint parameters. For example, consider a spherical joint. The Mathematica input would look as follows.

```
(* Define Joint Map Matrix *)
   H=Join [Identity Matrix[3],
          DiagonalMatrix [{0,0,0}]]
(* Name Joint Parameters *)
   y={y1,y2,y3}
(* Compute Configuration Matrix *)
   XXeuc [H,y]
(* Compute Kinematic Matrix *)
   GammaKin[H,y]
```

Typical results of such calculations are given in Table II.

D. Compound Kinematic Joints

Not all joints are simple kinematic joints. But in many cases it is possible to define the action of a joint in terms of a sequence of simple kinematic joints. We call such joints *compound kinematic joints*. In general, a compound joint is defined as a joint which can be characterized as the relative motion of a sequence of p reference frames such that relative motion between two successive frames is defined by a simple kinematic joint. Then each of the p simple joints is characterized by a joint map matrix $\mathcal{H}_i$ with r_i columns, a quasivelocity vector, β_i, of dimension r_i, a prameter vector, z_i, of dimension r_i, and a kinematic matrix $\Gamma_i(z_i)$. Thus if we define $z := [z_1 \cdots z_p]$ and $\beta := [\beta_1 \cdots \beta_p]$, we have the joint kinematics defined by

$$\dot{z} = \text{diag}[\Gamma_1(z_1), \ldots, \Gamma_p(z_p)]\beta \tag{25}$$

and, assuming the frames are indexed from the outermost, the overall joint configuration matrix is

$$X(z) = X_p(z_p)\cdots X_2(z_2)X_1(z_1). \tag{26}$$

Equations (25) and (26) provide the kinematic equations for compound joints.

TABLE II
COMPUTER GENERATED KINEMATICS ($\Gamma(\epsilon)$) FOR SOME SIMPLE JOINTS

H={0,0,1,0,0,0}, **revolute**

{1}

H={0,0,0,1,0,0}, **prismatic**

{1}

H={1,0,0,s,0,0}, **screw**

{1}

H={{0, 1}, {0, 0}, {1, 0}, {0, s}, {0, 0}, {0, 0}}, **simple universal-screw**

{{1, 0}, {0, Cos[z1]}}

H={{0, 0}, {1, 0}, {0, 1}, {0, 0}, {0, 0}, {0, 0}}, **simple universal**

{{1, 0}, {0, Cos[z1]}}

H={{1, 0, 0}, {0, 1, 0}, {0, 0, 1}, {0, 0, 0}, {0, 0, 0}, {0, 0, 0}}, **spherical**

{{1, Sin[y1] Tan[y2], Cos[y1] Tan[y2]},
{0, Cos[y1], -Sin[y1]},
{0, Sec[y2] Sin[y1], Cos[y1] Sec[y2]}}

Remark: In view of (26) and Corollary 1, a p-frame compound joint with joint map matrices $\mathcal{H}_i$, $i = 1, \ldots, p$, yields the same configuration manifold parameterization as a simple joint with joint map matrix $H = [\mathcal{H}_1 \cdots \mathcal{H}_p]$.

As we will see below, the overall joint map matrix is also required in order to assemble the dynamical equations for multibody systems. The required constructions are provided in the following proposition.

Proposition 3: Consider a compound joint composed of p simple joints with joint map matrices $\mathcal{H}_i = [h_i^1 \cdots h_i^{r_i}] \in R^{6\times r_i}$, $i = 1, \ldots, p$. Suppose $z := [z_1 \cdots z_p]$ and $\beta := [\beta_1 \cdots \beta_p]$ are the corresponding simple joint parameters and quasivelocities. Then the composite joint map matrix $H(z) \in R^{6\times(r_1+\cdots+r_p)}$ is given by the following construction:

$$H(z) := \begin{bmatrix} h_{11} & \cdots & h_{1r} \\ h_{21} & \cdots & h_{2r} \end{bmatrix} \tag{27}$$

where

$$\tilde{h}_{j1} = \mathcal{L}_{i-1}\tilde{h}_{i1}^j\mathcal{L}_{i-1}^T,\quad h_{j2} = \mathcal{L}_{i-1}\tilde{h}_{i1}^j\mathcal{R}_{i-1} + \mathcal{L}_{i-1}h_{i2}^j,$$
$$\text{for } i = 1, \ldots, p \quad\text{and}\quad j = 1, \ldots, r_i \tag{28a}$$
$$\mathcal{L}_i^T(z_i, \ldots, z_1) = L_i^T(z_i)\mathcal{L}_{i-1}^T(z_{i-1}, \ldots, z_1),\quad \mathcal{L}_0^T = I \tag{28b}$$
$$\mathcal{R}_i(z_i, \ldots, z_1) = L_i^T(z_i)\mathcal{R}_{i-1}(z_{i-1}, \ldots, z_1) + R_i,\quad \mathcal{R}_0 = 0. \tag{28c}$$

Proof: The overall joint velocity is

$$\dot{X} = \sum_{i=1}^{p}\sum_{j=1}^{r_i}\frac{\partial\chi}{\partial z_i^j}\dot{z}_i^j = \chi(z)p. \tag{29}$$

Notice that for each fixed $i \geq 2$,

$$\sum_{j=1}^{r_i}\frac{\partial\chi}{\partial z_i^j}\dot{z}_i^j = \chi_p(z_p)\cdots\chi_{i+1}(z_{i+1})\left\{\sum_{j=1}^{r_i}\frac{\partial\chi_i(z_i)}{\partial z_i^j}\dot{z}_i^j\right\}$$
$$\times\,\chi_{i-1}(z_{i-1})\ldots\chi_1(z_1). \tag{30}$$

But, as computed above for simple joints,

$$\sum_{j=1}^{r_i}\frac{\partial\chi_i(z_i)}{\partial z_i^j}\dot{z}_i^j = \chi_i(z_i)\begin{bmatrix}\tilde{\mathcal{H}}_{i1}\beta_i & \mathcal{H}_{i2}\beta_i \\ 0 & 0\end{bmatrix}. \tag{31}$$

Thus we have

$$\dot{X} = \begin{bmatrix}\tilde{\mathcal{H}}_{11}\beta_1 & \mathcal{H}_{12}\beta_1 \\ 0 & 0\end{bmatrix} + \sum_{i=2}^{p}\chi_p(z_p)\cdots\chi_i(z_i)$$
$$\times\begin{bmatrix}\tilde{\mathcal{H}}_{i1}\beta_i & \mathcal{H}_{i2}\beta_i \\ 0 & 0\end{bmatrix}\chi_{i-1}(z_{i-1})\cdots\chi_1(z_1) = \chi(z)p$$

or, premultiplying through by $\chi(z)$,

$$p = \begin{bmatrix}\tilde{\omega}_b & v_b \\ 0 & 0\end{bmatrix}$$
$$= \begin{bmatrix}\tilde{\mathcal{H}}_{11}\beta_1 & \mathcal{H}_{12}\beta_1 \\ 0 & 0\end{bmatrix} + \sum_{i=2}^{p}[\chi_{i-1}(z_{i-1})\cdots\chi_1(z_1)]^{-1}$$
$$\times\begin{bmatrix}\tilde{\mathcal{H}}_{i1}\beta_i & \mathcal{H}_{i2}\beta_i \\ 0 & 0\end{bmatrix}[\chi_{i-1}(z_{i-1})\cdots\chi_1(z_1)]. \tag{32}$$

This important relationship gives the body rates across the compound joint in terms of the joint quasivelocities.

Now, we can also write

$$\begin{bmatrix}\tilde{\mathcal{H}}_{i1}\beta_i & \mathcal{H}_{i2}\beta_i \\ 0 & 0\end{bmatrix} = \sum_{j=1}^{r_i}\begin{bmatrix}\tilde{h}_{i1}^j & h_{i2}^j \\ 0 & 0\end{bmatrix}\beta_i^j. \tag{33}$$

So that (32) can be written in the form

$$p = H(z)\beta \tag{34}$$

where $H(z)$ is constructed as follows:

$$\tilde{h}_{j1} = \mathcal{L}_{i-1}\tilde{h}_{i1}^j\mathcal{L}_{i-1}^T,\quad h_{j2} = \mathcal{L}_{i-1}\tilde{h}_{i1}^j\mathcal{R}_{i-1} + \mathcal{L}_{i-1}h_{i2}^j,$$
$$\text{for } i = 1, \ldots, p \text{ and } j = 1, \ldots, r_i.$$

In this case

$$\mathcal{L}_i^T(z_i, \ldots, z_1) = L_i^T(z_i)\mathcal{L}_{i-1}^T(z_{i-1}, \ldots, z_1),\quad \mathcal{L}_0^T = I$$
$$\mathcal{R}_i(z_i, \ldots, z_1) = L_i^T(z_i)\mathcal{R}_{i-1}(z_{i-1}, \ldots, z_1) + R_i,\quad \mathcal{R}_0 = 0.$$

□

Note that these equations differ from those of Proposition 2 only in that each z_i is a vector of dimension r_i rather than a scalar.

E. Computer Implementation

There are three constructions for compound joints: (1) assembly of the configuration matrix, $\chi(z)$, (2) assembly of the kinematic matrix $\Gamma(z)$, and (3) assembly of the joint map matrix $H(z)$. Correspondingly, we have three Mathematica functions to implement these calculations; `XXCmpnd`, `GamCmpnd` and `HCompnd`. The input to these functions consists of: a list of the p numbers $r_1, \ldots, r_p$, the matrix, $\mathcal{H}$, composed of p simple joint map matrices $\mathcal{H}_i \in R^{6\times r_i}$, $i = 1, \ldots, p$, and a list of $r = r_1 + \cdots + r_p$ names for the joint parameters.

A widely used example of a compound joint is the 3 degree of freedom universal joint. Such a joint is illustrated in the figure. This joint is composed of three elements and requires three frames to describe the composite motion. The relative motion between each of them involves one degree of freedom. In our terminology

$$\mathcal{H} = \begin{bmatrix}1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0\end{bmatrix}$$

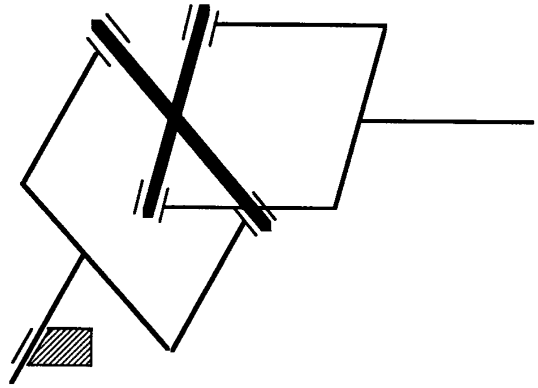

Fig. 1. Diagram of a 3 DOF universal joint. Note that the joint itself is composed of three bodies in addition to the fixed reference body.

```
(* Define Script-H *)
  H=Join[IdentityMatrix [3],
         DiagonalMatrix[{0,0,0}]]
(* Define dof Vector *)
  r={1,1,1}
(* Name Joint Parameters *)
  t={t1,t2,t3}
(* Compute Joint Map Matrix *)
  HCompnd[r,H,t]
(* Compute Configuration Matrix *)
  XXCmpnd[r,H,t]
(* Compute Kinematic Matrix *)
  GamCmpnd[r,H,t].
```

The results of this calculation are:

$$H = \begin{bmatrix} 1 & 0 & -\sin t2 \\ 0 & \cos t1 & \cos t2 \sin t1 \\ 0 & -\sin t1 & \cos t1 \cos t2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{35a}$$

see (35b) at the bottom of this page and

$$\Gamma = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{35c}$$

F. Remarks on Configuration Coordinates

The joint quasivelocities are naturally defined by the action of the joint. Joint configuration coordinates, however, are defined by the kinematic relation (16). While these equations formally define the coordinates (by defining z), they also provide a physical interpretation. Before examining some examples, note that $\Gamma(z)$ itself follows directly from the joint definition. Therefore to the extent that there is some freedom in specifying the joint parameters (the vector r and the matrix H), the user sets up the physical meaning of the coordinates z.

To see how this works, consider a general six degree of freedom joint (unconstrained 6 DOF relative motion) defined by:

```
r={6}; H=IdentityMatrix[6];
q={ax,ay,az,x,y,z};
p={wx,wy,wz,ux,uy,uz};
```

Consider this joint as depicting the relative motion of a body with respect to a space frame. The velocity transformation matrix Γ is:

$$\Gamma = \mathrm{diag}(\Gamma_1, \Gamma_2) \tag{36a}$$

$$\Gamma_1 = \begin{bmatrix} 1 & \sin ax \tan ay & \cos ax \tan ay \\ 0 & \cos ax & -\sin ax \\ 0 & \sec ay \sin ax & \cos ax \sec ay \end{bmatrix} \tag{36b}$$

see (36c) at the bottom of the page.

Inspection and comparison with standard results (e.g., [18]) reveals that the coordinates ax, ay, az are Euler angles in the 3-2-1 convention, and the coordinates x, y, z define the postion of the body frame relative to the space frame, as represented in the space frame. In other words, the quasivelocity vector (ux, uy, uz) corresponds to the body linear velocity in the body frame whereas the coordinate velocity $(\dot{x}, \dot{y}, \dot{z})$ represent the same body linear velocity in the space frame. By interchanging the first three columns of H, the resultant angle parameters again turn out to be Euler parameters, but in different conventions. If the columns in H corresponding to angles and linear displacements are interchanged, then the representation of the linear velocity and displacement will switch from space to body frame (or vice-versa).

Our parameterization is more flexible than the commonly used Denavit-Hartenberg (D-H) parameterization [19] because the model builder can, to some extent, control the physical meaning of the joint coordinates. Moreover, to use the D-H parameters, each joint must be treated as a compound sequence of one degree of freedom joints which eliminates the potentially substantial simplification of the dynamical equations afforded by Poincaré's form of Lagrange's equations. We will discuss this point below. The great advantage of the D-H parameterization is that the configuration matrix is of a fixed and specific form that is particularly easy to use when doing kinematic calculations by hand for systems involving revolute and prismatic joints. This, of course, was the overriding consideration at the time of publication of [19], but it is not an essential factor when using computer algebra constructions.

III. Poincaré's Equations

Our approach to multi-flex-body modeling is based on the Lagrangian framework. The Lagrangian dynamics for multibody sys-

$$X = \begin{bmatrix} \cos t2 \cos t3 & \cos t3 \sin t1 \sin t2 - \cos t1 \sin t3 & \cos t1 \cos t3 \sin t2 + \sin t1 \sin t3 & 0 \\ \cos t2 \sin t3 & \cos t1 \cos t3 + \sin t1 \sin t2 \sin t3 & -\cos t3 \sin t1 + \cos t1 \sin t2 \sin t3 & 0 \\ -\sin t2 & \cos t2 \sin t1 & \cos t1 \cos t2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{35b}$$

$$\Gamma_2 = \begin{bmatrix} \cos ay \cos az & \cos az \sin ax \sin ay - \cos ax \sin az & \cos ax \cos az \sin ay + \sin ax \sin az \\ \cos ay \sin az & \cos ax \cos az + \sin ax \sin ay \sin az & -\cos az \sin ax + \cos ax \sin ay \sin az \\ -\sin ay & \cos ay \sin ax & \cos ax \cos ay \end{bmatrix} \tag{36c}$$

tems are conveniently formulated using quasivelocities [2]–[4] which result in a system of equations often called Poincaré's equations. The method has been further developed using the recursive constructions introduced by Rodriguez and Jain [5]–[7] for serial chains of articulating bodies.

A. Hamilton's Principle and the Euler-Lagrange Equations

The formalism of Lagrangian dynamics begins with the identification of a configuration space, i.e., a set of points which form a manifold, M, and which are in a one-to-one correspondence with the possible physical configurations of the system of interest. The velocity at a point $q \in M$ is an element, $\dot{q}$, belonging to the tangent space to M at q, often denoted $T_q M$. The state space is defined as the union of tangent spaces at all points $q \in M$, the so-called tangent bundle TM. The evolution of the system in the state space is characterized by the definition of a Lagrangian $\mathcal{L}(q, \dot{q}) : TM \to R$ and use of Hamilton's principle of least action. This leads to the Euler-Lagrange equations

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = Q. \tag{37}$$

In the usual case we have $\mathcal{L}(q, \dot{q}) = \mathcal{T}(q, \dot{q}) - \mathcal{V}(q)$, where $\mathcal{T}$ and $\mathcal{V}$ are the kinetic energy and potential energy functions, respectively.

It is well known that in some cases it is easier to formulate the equations of motion in terms of velocity variables which can not be expressed as the time derivatives of any corresponding configuration coordinates. Such velocities are called quasivelocities. Quasivelocities are meaningful physical quantities—the angular velocity of a rigid body is a prime example. Such alternatives to Lagrange's equations were produced at the turn of the century (see, for example, [16] and [2]).

Let M be the m-dimensional configuration manifold for a Lagrangian system and suppose $v_1, \ldots, v_m$ constitute a system of m linearly independent vector fields on M. Then each commutator of pairs of vector fields can be expressed

$$[v_i, v_j] = \sum_{k=1}^{m} c_{ij}^{k}(q) v_k. \tag{38}$$

Suppose $q(t)$: $[t_1, t_2] \to M$ is a smooth path; then $\dot{q}(t)$ denotes the tangent vector to the path at the point $q(t) \in M$. Thus, we can always express $\dot{q}$ as a linear combination of the tangent vectors $v_i, i = 1, \ldots, m$

$$\dot{q} = V(q)p. \tag{39}$$

The variables p are called *quasivelocities.*

It is always possible to write the Lagrangian in terms of q and p using (39). Set $\hat{\mathcal{L}}(q, p) = \mathcal{L}(q, \dot{q})$. In terms of $\hat{\mathcal{L}}$ Lagrange's equations are attainable in the form given by the following result.

Proposition 4: Hamilton's principles leads to the equations of motion in terms of the coordinates q, p

$$\frac{d}{dt}\frac{\partial \hat{\mathcal{L}}}{\partial p_k} - \sum_{i,j=1}^{m} c_{jk}^{i} \frac{\partial \hat{\mathcal{L}}}{\partial p_i} p_j - \mathbf{v_k}(\hat{\mathcal{L}}) = Q^T v_k \tag{40}$$

or, equivalently,

$$\frac{d}{dt}\frac{\partial \hat{\mathcal{L}}}{\partial p} - \frac{\partial \hat{\mathcal{L}}}{\partial p} V^{-1} \sum_{j=1}^{m} p_j X_j - \frac{\partial \hat{\mathcal{L}}}{\partial q} V = Q^T V \tag{41}$$

where $V = [v_1\ v_2 \cdots v_m]$ and $X_j = [[v_j, v_1][v_j, v_2] \cdots [v_j, v_m]]$. $\mathbf{v_k}$ denotes the differential operator form of v_k.

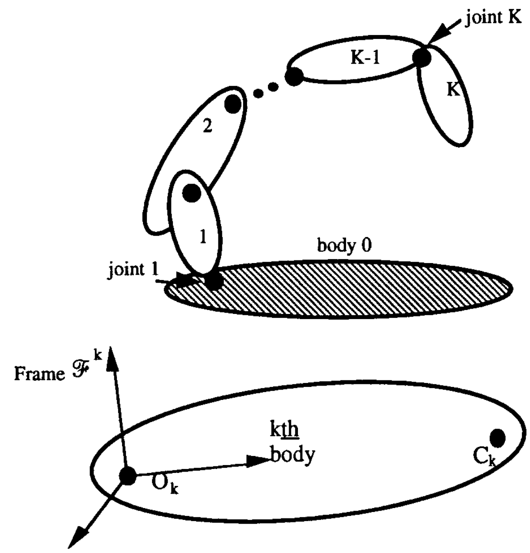

Fig. 2. A serial chain composed of $K + 1$ rigid bodies numbered 0 through K and K joints numbered 1 through K. On an arbitrary kth link the inboard and outboard joint hinge points are designated O_k and C_k. The body fixed reference frame $\mathcal{F}^k$ has its origin at O_k.

A proof of the proposition as stated here is given in [2]. Alternate derivations may be found in [4], [16], and [3].

Remark: If $V(q) = I$, then Poincaré's equations are Lagrange's equations. In this case the vector fields v_i are aligned with the configuration coordinates.

B. Kinetic Energy for Serial Chains of Rigid Bodies

The key issue in applying Lagrange's or Poincaré's equations to complex multibody dynamics is the formulation of the kinetic energy function and we focus on that construction. Suppose C is any point fixed in a rigid body. Rodriguez et al. [5]–[7] define the *spatial velocity at point C* of any body-fixed reference frame with origin at point C as $V_c = [\omega, v_c]$ where v_c is the velocity of point C and ω is the angular velocity of the body. Let O be another point in the same body and let r_{co} denote the location of C in the body frame with origin at O. Then the spatial velocity at point C is related to that at O by the relation

$$V_c = \phi(r_{co}) V_o \tag{42}$$

where

$$\phi(r_{co}) = \begin{bmatrix} I & 0 \\ -\tilde{r}_{co} & I \end{bmatrix}, \text{ and its adjoint } \phi^*(r_{co}) = \begin{bmatrix} I & \tilde{r}_{co} \\ 0 & I \end{bmatrix}. \tag{43}$$

Now, consider a serial chain composed of $K + 1$ rigid bodies connected by joints as illustrated in Fig. 2. The bodies are numbered 0 through K, with 0 denoting the base or reference body, which may represent any convenient inertial reference frame. The kth joint connects body $k-1$ at the point C_{k-1} with body k at the point O_k.

Let $\mathcal{F}^k$ denote a reference frame fixed in body k with origin at O_k. r_{co}^k denotes the vector from O_k to C_k in $\mathcal{F}^k$ and r^k denotes the vector from O_k to O_{k+1} in $\mathcal{F}^k$. We will use a coordinate specific notation in which vectors represented in $\mathcal{F}^i$ (or its tangent space) will be identified with a superscript "i". Coordinate free relations carry no superscript. The kth joint has n_k, $1 \le n_k \le 6$ degrees of freedom which can be characterized by n_k quasivelocities $\beta(k)$ and a joint map matrix $H(k) \in R^{6\times n_k}$ so that $V_{o_k} - V_{c_{k-1}} = H(k)\beta(k)$.

Rodriguez and his coworkers establish the recursive velocity relation which we write in coordinate specific notation

$$V^c(k) = \phi(r^c_{co}(k-1))V^c(k-1) + H^c(k)\beta^c(k). \tag{44}$$

Let us assume that $H(k)$ and $\beta(k)$ are specified in the frame $\mathcal{F}^k$ and $V(k-1)$ has been computed in the frame $\mathcal{F}^{k-1}$. Then it is convenient to compute $V(k)$ in the kth frame

$$\begin{aligned} V^k(k) = {} & \mathrm{diag}(L_{k-1,k}, L_{k-1,k})\phi(r^{k-1}_{co}(k-1))V^{k-1}(k-1) \\ & + H^k(k)\beta^k(k). \end{aligned} \tag{45}$$

If $V^0(0)$ is given, then (45) allows us to compute recursively, for $k = 1, \ldots, K$, the linear velocity of the origin of $\mathcal{F}^k$ and the angular velocity of $\mathcal{F}^k$, both represented in the coordinates of $\mathcal{F}^k$. In what follows we take $V^0(0) = 0$. Abusing notation somewhat, it is convenient to define

$$\phi(k, k-1) = \mathrm{diag}(L_{k-1,k}, L_{k-1,k})\phi(r^{k-1}_{co}(k-1)) \tag{46}$$

so that (45) can be written

$$\begin{aligned} V^k(k) &= \phi(k, k-1)V^{k-1}(k-1) + H^k(k)\beta^k(k), \\ & k = 1, \ldots, K, \quad V^0(0) = 0. \end{aligned} \tag{47}$$

It is necessary to define a spatial inertia tensor as well. Consider the kth rigid link and let $I_{cm}(k)$ denote the inertia tensor about the center of mass in coordinates $\mathcal{F}^k$, $m(k)$ denote the mass and $a(k)$ denote the position vector from the center of mass to an arbitrary point O. The spatial inertia about the center of mass, M_{cm}, and about O, M_o, are

$$\begin{aligned} M_{cm}(k) &= \begin{bmatrix} I_{cm} & 0 \\ 0 & mI \end{bmatrix}, \\ M_o(k) &= \phi^*(a)M_{cm}\phi(a) = \begin{bmatrix} I_o & m\tilde{a} \\ -m\tilde{a} & mI \end{bmatrix} \end{aligned} \tag{48}$$

where I_o is the inertia tensor about O.

The spatial velocity and spatial inertia matrix and, hence, the kinetic energy function for the entire chain can now be conveniently constructed. Let us define the chain spatial velocity and joint quasivelocity

$$\mathbf{V} = [V^t(1), \ldots, V^t(K)]^t, \quad \beta = [\beta^t(1), \ldots, \beta^t(K)]^t \tag{49}$$

so that we can write

$$\mathbf{V} = \Phi\mathbf{H}\beta, \tag{50}$$

where

$$\begin{aligned} \Phi &= \begin{bmatrix} I & 0 & \cdots & 0 \\ \phi(2,1) & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \phi(K,1) & \phi(K,2) & \cdots & I \end{bmatrix}, \\ \mathbf{H} &= \begin{bmatrix} H(1) & 0 & \cdots & 0 \\ 0 & H(2) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & H(K) \end{bmatrix} \\ \phi(i,j) &= \phi(i, i-1)\cdots\phi(j+1, j), \\ & i = 2, \ldots, K \quad \text{and} \quad j = 1, \ldots, K-1. \end{aligned} \tag{51}$$

The following result is easily verified.

Proposition 5: The kinetic energy function for the chain consisting of links 1 through K is

$$K.E._{\text{chain}} = \frac{1}{2}\beta^t\mathcal{M}\beta \tag{52a}$$

where the chain inertia matrix is

$$\mathcal{M} = \mathbf{H}^T\Phi^T M\Phi\mathbf{H}, \quad M = \mathrm{diag}(M_o(1), \ldots, M_o(K)). \tag{52b}$$

Remark on the Structure of Poincaré's Equations: The above definitions and constructions provide the kinetic energy function in the form $\tilde{\mathcal{T}}(q,p) = (1/2)p^T\mathcal{M}(q)p$. Hence, we reduce (41) to the form:

$$\mathcal{M}(q)\dot{p} + \mathcal{C}(q,p)p + \mathcal{F}(q) = Q_p \tag{53a}$$

where

$$\begin{aligned} \mathcal{C}(q,p) = {} & -\left[\frac{\partial(\mathcal{M}p)}{\partial q}V\right] + \frac{1}{2}\left[\frac{\partial(\mathcal{M}p)}{\partial q}V\right]^T \\ & + \left[\sum_{j=1}^{m} p_j X_j^T\right]V^{-T}\mathcal{M} \end{aligned} \tag{53b}$$

$$\mathcal{F}(q) = V^T(q)\frac{\partial\mathcal{V}(q)}{\partial q^T}, \quad Q_p = V^T(q)Q. \tag{53c}$$

Notice that Q_p denotes the generalized forces represented in the p-coordinate frame whereas Q denotes the generalized forces in the $\dot{q}$-coordinate frame (aligned with q).

Remark on Computations: The key point to be noted is that the matrix Φ (and hence the product $\Phi\mathbf{H}$) can be recursively computed. Thus, we can compute the spatial velocity of any or all of the bodies via (49) and the inertia matrix using (52b). Once this is done, we compute $\mathcal{C}(q,p)$, $\mathcal{F}(q)$ and Q_p explicitly using (53b and 53c), assuming that the potential energy function $\mathcal{V}(q)$ and the generalized force vector Q are available. In general, both $\mathcal{V}$ and Q are defined in terms of coordinates and velocities (in the case of Q) other than the configuration coordinates q and the quasivelocities p. Thus, it is necessary to develop any transformations required to obtain $\mathcal{V}$ and Q in terms of q and p. We cannot give a complete discussion of this process here but note that velocity transformations are recursively constructed using relations like (47) or (50), and coordinate transformations are built up from the usual sequential multiplications of configuration matrices. Assembly of the system gravitational potential energy and end effector position and orientation, needed below, require constructions of this type.

Remark on the use of Poincaré versus Lagrange Equations: Notice that the kinetic energy can be expressed in terms of $\dot{q}$ rather than p, $\mathcal{T}(q,\dot{q}) = (1/2)\dot{q}^T\{V(q)^{-T}\mathcal{M}(q)V(q)^{-1}\}\dot{q}$, and hence we have the essential data to construct Lagrange's equations rather than Poincaré's equations. However, Poincaré's equations may have important advantages. An obvious and practical one is the relative simplicity of the inertia matrix. However, there is an important theoretical consideration as well. Lagrange's equations fundamentally constitute a local representation whenever local coordinates are introduced, whereas Poincaré's equations may still admit a global description of the dynamics. This is easily seen by comparing the Lagrange and Poincaré formulations for the dynamics of a rotating rigid body. We do this below.

C. Computer Implementation

To formulate the equations of motion for a chain of rigid bodies several Mathematica functions have been implemented. We describe three principle functions: `RgdChn`, `Cmat`, and `PoincFunc`. `RgdChn` generates all of the kinematic relations and the chain inertia matrix. It has three inputs: a list of joint data, a list of body data, and a list of coordinate names. Each joint is defined by two pieces of data as described above; a p-vector $r_1, \ldots, r_p$ and a matrix $\mathcal{H} \in R^{6\times r}$ with $r = r_1 + \cdots + r_p$. The joint data list is a list of such pairs. Each body is characterized by four pieces of data; the location of the center of mass in a body frame with origin at the inboard joint, the location of the outboard joint in the same frame, the body mass, and the body inertia tensor about the center of mass. The body data list is a list of such quadruples. `RgdChn` returns a list of three items: (1) a list of square matrices which are the diagonal elements of the block diagonal kinematic matrix, (2) a list of 4 × 4 matrices which are the Euclidean configuration matrices for the joints, and (3) the chain inertia matrix. The function `Cmat` receives as input the chain inertia matrix, M, the list of diagonal matrices of the kinematic matrix, V, the list of coordinate names, q and a list of velocity names, p. It returns the matrix $\mathcal{C}(q,p)$. The function `PoincFunc` receives M, V, q, p and also the potential energy function, $\mathcal{V}(q)$, and a vector of generalized forces Q_p. It returns the vector function $F = \mathcal{C}(q,p)p + \mathcal{F}(q) - Q_p$, thereby completing Poincaré's equations.

As an example of the application of these functions let us consider a thin disk free to rotate about its center of mass in space without any external or gravitational forces. The single joint defining relative motion between the space frame and the body frame is considered as a simple spherical joint.

```
(* Rigid Body Example–Thin Disk *)
(* Spherical Joint *)
  r1={3};
  H1= Join[IdentityMatrix[3],
  DiagonalMatrix[{0,0,0}]];
  JointLst={{r1,H1}};
(* Body—thin disk *)
  m1=5;R1=2;I1=DiagonalMatrix[ {(1/4)
  *m1*R1^2,(1/4)*m1*R1^2,(1/2)
  *m1*R1^2 } ];
  cm1= {0,0,0};oc1={1,0,0};
  BodyLst= {{cm1,oc1,m1,I1}};
(* Name Coordinates *)
  q={t1,t2,t3};
(* Compute Matrices: Kinematic,
   Configuration, Inertia *)
  {V,X,M}=RgdChn[JointLst,BodyLst,q]
(* Name Quasi-Velocities *)
  p={w1,w2,w3};
(* Compute C-Matrix *)
  GG=Cmat[M,V,q,p]
```

We summarize the results as follows: $X(t)$ is given by (35b),

$$\Gamma(t) = \begin{bmatrix} 1 & \sin t1 \tan t2 & \cos t1 \tan t2 \\ 0 & \cos t1 & -\sin t1 \\ 0 & \sec t2 \sin t1 & \cos t1 \sec t2 \end{bmatrix}, \tag{54a}$$

$$\mathcal{M}(t) = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 10 \end{bmatrix}, \quad F(t,w) = \mathcal{C}(q,p)p = \begin{bmatrix} -5w2w3 \\ 5w1w3 \\ 0 \end{bmatrix}. \tag{54b}$$

Poincaré's equations are recognizable as Euler's equations.

It is interesting to repeat this calculation with the simple spherical joint replaced by a compound 3 dof universal joint. The only change required in the above Mathematica code is to replace the definition of `r1={3}` by `r1={1,1,1}`. As noted above, the parameterization of the configuration of the rigid body is the same as that of the simple joint, i.e., $X(t)$ is unchanged and $\Gamma(t) = I_3$. The other relevant results are as follows

$$\mathcal{M}(t) = \begin{bmatrix} 5 & 0 & -5\sin t2 \\ 0 & 5/2(3-\cos 2t1) & -5/2\cos t2 \sin 2t1 \\ -5\sin t2 & -5/2\cos t2\sin 2t1 & \begin{array}{c} 10\cos t1^2\cos t2^2 \\ +5\cos t2^2 \sin t1^2 \\ +5\sin t2 \end{array} \end{bmatrix} \tag{55a}$$

$$\begin{aligned} F_1(t,u) &= u3(-5u2\cos 2t1\cos t2 - 5u3\cos t2^2 \sin 2t1)/2 \\ &\quad + u2(5u3\cos t2 + (-5u3\cos 2t1 \cos t2 \\ &\quad + 5u2\sin 2t1)/2) \\ F_2(t,u) &= u1(-5u3\cos t2/2 + 5u3\cos 2t1\cos t2 - 5u2\sin 2t1) \\ &\quad - 5u2u3\sin 2t1 \sin t2/4 + (u3(-5u1\cos t2 \\ &\quad + 5u2\sin 2t1\sin t2/2 - 5u3\cos t1^2 \sin 2t2))/2 \\ F_3(t,u) &= u1(5u2\cos 2t1\cos t2 + 5u3\cos t2^2\sin 2t1) \\ &\quad + u2(5u1\cos t2 - 5u2\sin 2t1\sin t2/2 \\ &\quad + 5u3\cos t1^2 \sin 2t2). \end{aligned} \tag{55b}$$

These equations are, of course, Lagrange's equations—a consequence of the fact that the joint formulation commits us to velocity coordinates aligned with the configuration coordinates. The simplicity of the kinematic matrix is more than offset by the complexity of the dynamical equations. Notice also that the dynamical equations in the previous case, defined by (54b), are independent of the configuration parameters. They are globally valid equations, whereas the latter are not.

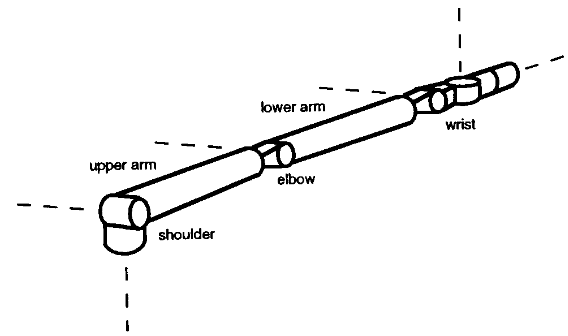

Fig. 3. The six degrees of freedom arm used in the example.

IV. Example

As a more complete example of the methods described above, we consider a six degree of freedom robot arm. The system is composed of three joints and three bodies as illustrated in Fig. 3.

The following Mathematica program constructs all of the kinematic relationships and Poincaré's equations.

```
(* joint 1—compound shoulder *)
  r1={1,1};
  H1={{0,0}, {1,0}, {0,1}, {0,0}, {0,0},
      {0,0}};
(* Joint 2—revolute elbow *)
  r2={1};
  H2={{0}, {1}, {0}, {0}, {0}, {0}};
(* Joint 2—compound wrist *)
  r3={1,1,1};
  H3={{0,0,1}, {1,0,0}, {0,1,0}, {0,0,0},
      {0,0,0}, {0,0,0}};
JointLst={{r1,H1}, {r2,H2}, {r3,H3}};
```

```
(* Body 1—slender bar upper arm *)
  m1=1;L1=3;I1=DiagonalMatrix[{(1/12)*m1
            *L1^2, (1/12)*m1*L1^2,0}];
  cm1={0,0,L1/2};oc1={0,0,L1};
(* Body 2—slender bar lower arm *)
  m2=1;L2=3;I2=DiagonalMatrix[{(1/12)*m2
            *L2^2, (1/12)*m2*L2^2,0}];
  cm2={0,0,L2/2}; oc2={0,0,L2};
(* Body 3—cylindrical gripper *)
  m3=.5; L3=.7; R3=.35;
  I3 = DiagonalMatrix[{(1/12)*m3*
      (L3^2+3*R3^2),(1/12)*m3*
      (L3^2+3*R3^2),(1/12)*m3*R3^2}];
  cm3={0,0,L3/2}; oc3={0,0,L3};
BodyLst={{cm1,oc1,m1,I1},{cm2,oc2,m2,I2},
      {cm3,oc3,m3,I3}};
(* coordinates *)
q={s1,s2,e,t1,t2,t3};
{V,X,M}=RgdChn[JointLst,BodyLst,q]
XEnd=XEndEf[BodyLst,X,q]
(* quasivelocities *)
p={w1,w2,u1,v1,v2,v3};
PE=GravPotChn[BodyLst,X,q]
Q={T1,T2,T3,T4,T5,T6};
F=PoincFunc[M,V,PE,Q,p,q]
```

Two new functions are used. XEndEf computes the configuration matrix of the end effector as a function of the joint parameters. GravPotChn computes the gravitational potential energy function. Both functions require the same three pieces of information as input: the list of body data, the list of joint configuration matrices, and the list of joint parameter names. In this example, the function XEndEf returns the configuration matrix of the end effector relative to the space frame:

```
{{Cos[t1]Cos[t2](Cos[e]Cos[s1]Cos[s2]
-Cos[s2]Sin[e]Sin[s1])-
Sin[s2](Cos[t1]Cos[t3]
Sin[t2]+Sin[t1]Sin[t3])
+(Cos[s1]Cos[s2]Sin[e]+Cos[e]
Cos[s2]Sin[s1])(-(Cos[t3]Sin[t1])
+Cos[t1]Sin[t2]Sin[t3]),

-(Cos[t2]Cos[t3]Sin[s2])
-(Cos[e]Cos[s1]Cos[s2]
-Cos[s2]Sin[e]Sin[s1])Sin[t2]
+Cos[t2](Cos[s1]Cos[s2]Sin[e]
+Cos[e]Cos[s2]Sin[s1])Sin[t3],

Cos[t2](Cos[e]Cos[s1]Cos[s2]
-Cos[s2]Sin[e]Sin[s1])Sin[t1]-
Sin[s2](Cos[t3]Sin[t1]Sin[t2]
-Cos[t1]Sin[t3])+(Cos[s1]Cos[s2]Sin[e]
+Cos[e]Cos[s2]Sin[s1])(Cos[t1]Cos[t3]
+Sin[t1]Sin[t2]Sin[t3]),

3Cos[s2]Sin[s1]+3(Cos[s1]Cos[s2]Sin[e]
+Cos[e]Cos[s2]Sin[s1])+0.7(Cos[t2]
(Cos[e]Cos[s1]Cos[s2]
-Cos[s2]Sin[e]Sin[s1])Sin[t1]-
Sin[s2](Cos[t3]Sin[t1]Sin[t2]
-Cos[t1]Sin[t3])+(Cos[s1]Cos[s2]Sin[e]
+Cos[e]Cos[s2]Sin[s1])(Cos[t1]Cos[t3]
+Sin[t1]Sin[t2]Sin[t3]))},

{Cos[t1]Cos[t2](Cos[e]Cos[s1]Sin[s2]-
Sin[e]Sin[s1]Sin[s2])+Cos[s2]
(Cos[t1]Cos[t3]Sin[t2]+Sin[t1]Sin[t3])
+(Cos[s1]Sin[e]Sin[s2]+Cos[e]
Sin[s1]Sin[s2])(-(Cos[t3]Sin[t1])
+Cos[t1]Sin[t2]Sin[t3]),

Cos[s2]Cos[t2]Cos[t3]
-(Cos[e]Cos[s1]Sin[s2]
-Sin[e]Sin[s1]Sin[s2])Sin[t2]
+Cos[t2](Cos[s1]Sin[e]Sin[s2]
+Cos[e]Sin[s1]Sin[s2])Sin[t3],

Cos[t2](Cos[e]Cos[s1]Sin[s2]
-Sin[e]Sin[s1]Sin[s2])Sin[t1]
+Cos[s2](Cos[t3]Sin[t1]Sin[t2]
-Cos[t1]Sin[t3])+(Cos[s1]Sin[e]Sin[s2]
+Cos[e]Sin[s1]Sin[s2])(Cos[t1]Cos[t3]
+Sin[t1]Sin[t2]Sin[t3]),

3Sin[s1]Sin[s2]+3(Cos[s1]Sin[e]Sin[s2]
+Cos[e]Sin[s1]Sin[s2])
+0.7(Cos[t2](Cos[e]Cos[s1]Sin[s2]
-Sin[e]Sin[s1]Sin[s2])Sin[t1]
+Cos[s2](Cos[t3]Sin[t1]Sin[t2]
-Cos[t1]Sin[t3])+(Cos[s1]Sin[e]Sin[s2]
+Cos[e]Sin[s1]Sin[s2])(Cos[t1]Cos[t3]
+Sin[t1]Sin[t2]Sin[t3]))},

{Cos[t1]Cos[t2](-(Cos[s1]Sin[e])
-Cos[e]Sin[s1])+(Cos[e]Cos[s1]
-Sin[e]Sin[s1])(-(Cos[t3]Sin[t1])
+Cos[t1]Sin[t2]Sin[t3]),

-((-(Cos[s1]Sin[e])
-Cos[e]Sin[s1])Sin[t2])
+Cos[t2](Cos[e]Cos[s1]
-Sin[e]Sin[s1])Sin[t3],

Cos[t2](-(Cos[s1]Sin[e])
-Cos[e]Sin[s1])Sin[t1]
+(Cos[e]Cos[s1]
-Sin[e]Sin[s1])(Cos[t1]Cos[t3]+
Sin[t1]Sin[t2]Sin[t3]),

3Cos[s1]+3(Cos[e]Cos[s1]-Sin[e]Sin[s1])
+0.7(Cos[t2](-(Cos[s1]Sin[e])
-Cos[e]Sin[s1])Sin[t1]+(Cos[e]Cos[s1]
-Sin[e]Sin[s1])(Cos[t1]Cos[t3]
+Sin[t1]Sin[t2]Sin[t3])) },

{0,0,0,1}}
```

The function GravPotChn returns the potential energy function:

```
3Cos[s1]/2+3(Cos[e]Cos[s1]
-Sin[e]Sin[s1])/2
+0.35(Cos[t2](-(Cos[s1]Sin[e])
-Cos[e]Sin[s1])Sin[t1]
+(Cos[e]Cos[s1]-Sin[e]Sin[s1])
(Cos[t1]Cos[t3]+Sin[t1]Sin[t2]Sin[t3]))
```

The dynamical equations for this system are extremely complex and lengthy. Because of this we will not exhibit them here. Once again, however, a significant simplification is obtained by replacing the compound joints (joints 1 and 3) with their corresponding simple joint representations. This is accomplished by redefining $r1$, $r2$, by $r1 = \{2\}$ and $r3 = \{3\}$ in the above Mathematica program. As a measure of the reduction achieved we note that the inertia matrix so obtained is only about 25% as large as the compound case in terms of the length of the total mathematical expression. As an illustration we list only the last (6th) row of the inertia matrix for each case.

Last row of inertia matrix, compound joint representation

```
{1.05Cos[s1/2]^2Cos[t1]Cos[t2]Sin[s2]
+0.0969792Cos[e+s1]
 Cos[t1]Cos[t2]Sin[s2]
+0.00510417Cos[t2]Sin[e+s1]
 Sin[s2]Sin[t1]
-0.0969792Cos[s2]Sin[t2]
-1.05Cos[e + s1/2]Cos[s1/2]Cos[s2]Sin[t2],
-0.175Cos[t1]Cos[t2](6Cos[s1/2]
 Cos[s2]Sin[s1]
+3Sin[2s1]/2)
+0.0969792Cos[t1]Cos[t2]
-(-(Cos[e+s1]Cos[s2]Sin[s1])
-Cos[s1]Sin[e+s1])
+0.00510417Cos[t2](Cos[s1]Cos[e+s1]
-Cos[s2]Sin[s1]Sin[e+s1])Sin[t1]
-0.0969792Sin[s1]Sin[s2]Sin[t2]
-1.05Cos[e + s1/2]Cos[s1/2]
 Sin[s1]Sin[s2]Sin[t2],
-0.0969792Sin[t2]-0.525Cos[e]Sin[t2],
-0.0969792Sin[t2],
-0.091875Cos[t1]Cos[t2]Sin[t1],
 0.0969792Cos[t1]^2Cos[t2]^2+
 0.00510417Cos[t2]^2Sin[t1]^2+0.0969792
 Sin[t2]^2}
```

Last row of inertia matrix, simple joint representation

```
{1.05Cos[s1/2]^2Sin[s2]
+0.0969792Cos[e+s1]Sin[s2],
-0.525Sin[s1]-0.0969792Sin[e+s1],
 0,0,0,0.0969792}
```

V. Conclusion

In this paper, we have presented new algorithms for constructing models of joint kinematics and show how these algorithms can be used for deriving dynamical models for rigid multi-body chains in the form of Poincaré's equations. Computer software has been described which implements these constructions in the symbolic manipulation language Mathematica.

The joint representations are general in that they accommodate a large class of joints, and flexible in that the modeler has some freedom in setting up the physical meaning of joint coordinates when dealing with multi-degree of freedom joints. The computations themselves are recursive and efficient—by which we mean that the execution times are reasonable and that they produce recognizable and compact expressions when applied to standard joints.

Many multi-degree of freedom joints are physically realized as a sequence of one degree of freedom joints—a subclass of what we call compound joints. When joints are modeled in this way Poincaré's equations are Lagrange's equations. Any compound joint is (locally) kinematically equivalent to a simple joint and in the sense that both joint descriptions induce the same parameterization of the configuration matrix. We show that when a compound joint is represented by its equivalent simple joint, the resulting Poincaré's equations can be much less complex than Lagrange's equation. This is simply a result of the well known fact that such a simplification generally accompanies the use of quasivelocities.

The principle contributions of the paper are new algorithms for constructing joint representations and their application to the computer derivation of the equations of motion. These computations are not merely of theoretical interest. They form the basis of a comprehensive software package for modeling multibody systems with tree-like structure and composed of rigid and flexible links [20]. Using a MacIntosh Quadra and 486 PC, systems of modest scale have been investigated. That includes: derivation of the equations of motion, generation of C-code, compilation and simulation. The largest system considered to date is a 15 degree of freedom tracked vehicle with 10 road wheels and a flexible hull, as reported in [20]. Current work focuses on integrating this modeling tool with nonlinear control design software described in [1].

References

[1] G. L. Blankenship, R. Ghanadan, H. G. Kwatny and V. Polyakov, "Modeling and design tools for control of multibody systems," in *Proc. ASME Winter Annu. Meet.: High Performance Computing in Vehicle Systems*, 1993.

[2] V. I. Arnold, V. V. Kozlov, and A. I. Neishtadt, *Mathematical Aspects of Classical and Celestial Mechanics* (Encyclopedia of Mathematical Sciences), V. I. Arnold, Ed. Heidelberg: Springer-Verlag, 1988, vol. 3.

[3] N. G. Chetaev, *Theoretical Mechanics.* New York: Springer-Verlag, 1989.

[4] L. Meirovitch, *Methods of Analytical Dynamics.* New York: McGraw-Hill, 1970

[5] A. Jain, "Unified formulation of dynamics for serial rigid multibody systems," *AIAA J. Guidance, Cont. and Dynamics*, vol. 14, no. 3, pp. 531–542, 1991.

[6] A. Jain and G. Rodriguez, "Recursive flexible multibody system dynamics using spatial operators," *AIAA J. Guidance, Cont. and Dynamics*, vol. 15, no. 6, pp. 1453–1466, 1992.

[7] G. Rodriguez, "Kalman filtering, smoothing and recursive robot arm forward and inverse dynamics," *IEEE Trans. Robot. and Automat.*, vol. RA-3, no. 6, pp. 624–639, 1987.

[8] W. H. Bennett, H. G. Kwatny, and M. Baek, "Nonlinear dynamics and control of articulated flexible spacecraft: Application to SSF/MRMS," *AIAA J. Guidance, Cont. and Dynamics*, vol. 17, no. 1, pp. 38–47, 1994.

[9] T. J. Tarn, A. K. Bejzky, G. T. Marth, and A. K. Ramadorai, "Performance comparison of four manipulator servo schemes," *Cont. Syst.*, vol. 13, no. 1, pp. 22–29, 1993.

[10] M. A. Scott, M. G. Gilbert, and M. E. Demeo, "Active vibration damping of the space shuttle remote manipulator system," *AIAA J. Guidance, Cont. and Dynamics*, vol. 16, no. 2, pp. 275–280, 1993.

[11] E. J. Haug, Ed. *Computer Aided Analysis and Optimization of Mechanical System Dynamics (NATO ASI).* Springer-Verlag: Berlin, 1984, vol. 59.

[12] J. V. Ramakrishnan and R. P. Singh, "Computer aided design environment for the design of multi-body flexible structures," in *Proc. AIAA Guidance, Navigation and Cont.*, 1989, pp. 749–757.

[13] M. C. Leu and N. Hemati, "Automated symbolic derivation of dynamic equations for robotic manipulators," *J. Dynamic Syst., Measurement and Cont.*, vol. 108, pp. 172–179, Sept. 1986

[14] S. Cetinkunt and B. Ittoop, "Computer-automated symbolic modeling of dynamics of robotic manipulators with flexible links," *IEEE Trans. Robot. and Automat.*, vol. 8, no. 1, pp. 94–105, 1992.

[15] P. J. Olver, *Applications of Lie Groups to Differential Equations.* , New York: Springer-Verlag, 1986.

[16] J. I. Neimark and N. A. Fufaev, *Dynamics of Nonholonomic Systems* (Translations of Mathematical Monographs) Providence: American Mathematical Society, 1972, vol. 33.

[17] R. M. Rosenberg, *Analytical Dynamics of Discrete Systems.* New York: Plenum Press, 1977.

[18] H. Goldstein, *Classical Mechanics.* Reading: Addison-Wesley, 1980.

[19] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower pair mechanisms based on matrices," *ASME J. Appl. Mechan.*, vol. 22, pp. 215–221, June 1955.

[20] C. LaVigna, H. G. Kwatny, and G. L. Blankenship, "Flexible multibody dynamical analysis system," Phase I Final Report, Contract no. DAAE07-93-CR022, Techno-Sciences, Inc., 1993.