EFFICIENT DYNAMIC SIMULATION
FOR MULTIPLE CHAIN ROBOTIC MECHANISMS

Kathryn W. Lilly and David E. Orin

Department of Electrical Engineering
The Ohio State University
Columbus, OH 43210

Abstract

An efficient $O(mN)$ algorithm for dynamic simulation of simple closed-chain robotic mechanisms will be presented in this paper, where m is the number of chains, and N is the number of degrees of freedom for each chain. It is based on computation of the operational space inertia matrix (6×6) for each chain as seen by the body, load, or object. Also, computation of the chain dynamics, when opened at one end, is required, and the most efficient algorithm is used for this purpose. Parallel implementation of the dynamics for each chain results in an $O(N) + O(\log_2 m + 1)$ algorithm.

I. Introduction

Recently, there has been an increasing interest in robotic systems with multiple chains forming simple closed kinematic loops. Such systems of interest in space robotics applications include multilegged vehicles, multiple manipulators, and dexterous hands. Each is characterized by multiple chains of links (legs, arms, or fingers) in support of a body, load, or object. Real-time simulation of these systems is important for remote operation, but difficult to achieve at present. An even greater challenge to the computational engineer is that of *super-real-time simulation*, that is, planning seconds of motion in milliseconds. This has been shown to be of value in the control of a multilegged vehicle when predicting the action of the present control to ensure safety and stability along a planned trajectory.

The fundamental goal of this paper is the development of an efficient algorithm for the dynamic simulation of the time-varying topological systems discussed above. Previous researchers have presented algorithms for these and similar configurations based on equation augmentation [1], constraint propagation [2], and recursive computation [3,4], but these methods are often difficult to apply and/or computationally inefficient. The new simulation algorithm derived here makes use of efficient computations for the individual supporting chains to produce an efficient simulation method for the complete robot system. The dynamic properties of each chain are described in a simple, physically understandable manner, which facilitates the straightforward analysis of the combined dynamics of the entire mechanism.

Multiple chain robotic systems can take many forms, some of them quite complex. Simple closed-chain mechanisms are a subset of multiple chain systems with specific structural characteristics. The structure of a simple closed-chain mechanism is characterized by m actuated chains which support a single common reference member [1]. A supporting chain is identified as an independent functional unit in the closed chain system which has two ends, each terminated by a single link. Each chain may have an arbitrary number of links and degrees of freedom, and closed kinematic loops within a chain are permitted. The removal of the reference member breaks the closed loops formed by the multiple chains.
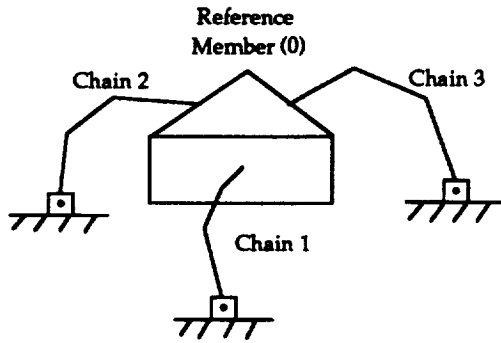
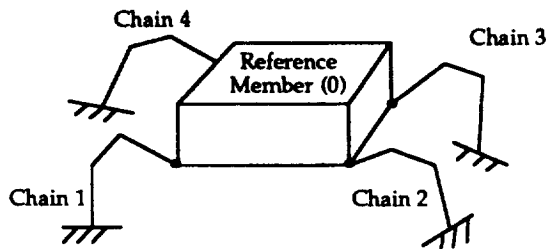Figure 1: Example of a Type 0 Simple Closed-Chain Mechanism

Figure 2: Example of a Type 1 Simple Closed-Chain Mechanism

There are two basic types of simple closed-chain mechanisms called Type 0 and Type 1, respectively [1]. These two types are defined based on the nature of the interactions which occur between the links of each chain and the reference member or support surface. Figure 1 illustrates a typical Type 0 mechanism which may be used to model multiple manipulators or dexterous hands. Note that the support surface, shown here as a fixed inertial frame for a multiple manipulator configuration, might also represent the moving "palm" of a dexterous hand. In either case, for a Type 0 mechanism, the base link of each chain is connected to the support surface by an actuated joint structure, while the last link interacts with the reference member through an unpowered contact. Figure 2 illustrates a Type 1 simple closed-chain mechanism which may be used to model multilegged vehicles. For a Type 1 mechanism, the last link of each chain interacts with the support surface through an unpowered contact, while the base link is connected to the reference member by an actuated joint structure. For both Type 0 and Type 1 mechanisms, the reference member (object, load, or body) is numbered 0, while the chains are numbered arbitrarily from 1 to m. Chain k ($k = 1,\ldots,m$) has N_k degrees of freedom, where N_k may be less than, equal to, or greater than 6.

In order to apply the same algorithm to both types in this work, the support surface will be considered to act as the "base" of each chain. We will refer to the terminal link which interacts with the support surface as link 1, and the terminal link which interacts with the reference member will be called the last link or end effector (link N). The far end of link N is the "tip" of the chain. The interactions and connections which occur between bodies or links in the system (including those at the support surface and at the reference member) will be described using the general joint model of [5,6]. This

74

includes both powered joint structures and unpowered contacts. The motion of the support surface is assumed to be known.

In this paper, an $O(mN)$ recursive algorithm for the dynamic simulation of simple closed-chain mechanisms is derived for m chains with N degrees of freedom each. The algorithm is based on the efficient computation of the (6 × 6) operational space inertia matrix [7] for each chain as seen by the body, load, or object. The operational space inertia matrix, Λ, may be used to obtain the net effect of the chain dynamics at its tip. The computation of the chain dynamics when the chain is open at one end is also required, and the most efficient algorithm is used for this purpose. Given $O(N)$ algorithms for these two fundamental computations for each chain [4,5,8,9], an $O(mN)$ algorithm for the simulation of the entire multiple chain system is formulated.

In the next section, the notational and modelling conventions used in the formulation of the new algorithm are summarized. In the third section, the dynamic properties of the individual supporting chains and the common reference member are discussed, and the appropriate dynamic equations are developed. The operational space inertia matrix and the open-chain dynamics of each chain are of special significance in this discussion. In the fourth section, the $O(mN)$ dynamic simulation algorithm for simple closed-chain mechanisms is derived. The final algorithm is presented as a series of five steps, which are summarized in a convenient tabular form. The computational requirements of the new algorithm, including parallel implementation considerations, are presented in the fifth section. Finally, the results of this work are summarized and some overall conclusions are given in the final section.

II. Notation

Many of the notational conventions used in this paper are based on concepts introduced by Roberson and Schwertassek in [6] and used by Brandl, Johanni, and Otter in [5]. They are similar in many ways to those described by Featherstone in [9] for robot dynamics, although there are a few minor differences. As in each of these, spatial notation will be used to develop the dynamic equations for the chains and reference member. With spatial notation, velocity, acceleration, and force vectors are all 6 × 1 column vectors, where each incorporates the appropriate linear and angular components. In this paper, the spatial velocity of the reference member, v_0, is written:

$$\mathbf{v}_0 = [\ (\omega_0)_x \quad (\omega_0)_y \quad (\omega_0)_z \quad (v_0)_x \quad (v_0)_y \quad (v_0)_z\ ]^T, \tag{1}$$

where $(\omega_0)_x$, $(\omega_0)_y$, and $(\omega_0)_z$ are the components of the angular velocity of the reference member about $\hat{x}$, $\hat{y}$, and $\hat{z}$, respectively, usually resolved in the reference member frame (frame 0) or an inertial coordinate system. The three components, $(v_0)_x$, $(v_0)_y$, and $(v_0)_z$, represent the linear velocity of the coordinate origin of frame 0. Similarly, the spatial acceleration of the reference member is expressed as:

$$\mathbf{a}_0 = [\ (\alpha_0)_x \quad (\alpha_0)_y \quad (\alpha_0)_z \quad (a_0)_x \quad (a_0)_y \quad (a_0)_z\ ]^T, \tag{2}$$

where the individual components now correspond to resolved angular and linear acceleration vectors. Spatial force vectors have a corresponding structure:

$$\mathbf{f}_k = [\ (n_k)_x \quad (n_k)_y \quad (n_k)_z \quad (f_k)_x \quad (f_k)_y \quad (f_k)_z\ ]^T, \tag{3}$$

where, in this case, $\mathbf{f}_k$ will be used to represent the spatial force exerted on the reference member by chain k. The first three components, $(n_k)_x$, $(n_k)_y$, and $(n_k)_z$, represent the elements of a three-dimensional moment vector, while $(f_k)_x$, $(f_k)_y$, and $(f_k)_z$ are the elements of a three-dimensional force vector.

In general, the transformation of a spatial velocity or acceleration vector from one coordinate system to another one may be accomplished by the following spatial multiplication [9]:

$$^j\mathbf{p} = {}^j\mathbf{X}_i {}^i\mathbf{p},\tag{4}$$

where $^i\mathbf{p}$ is the vector expressed with respect to the ith coordinate system, $^j\mathbf{p}$ is the same vector expressed with respect to the jth coordinate system, and $^j\mathbf{X}_i$ is the 6×6 spatial transformation matrix. This spatial transformation is defined as follows:

$$^j\mathbf{X}_i = \begin{bmatrix} {}^j\mathbf{A}_i & \mathbf{0} \\ {}^j\mathbf{A}_i \tilde{\mathbf{b}}_j^T & {}^j\mathbf{A}_i \end{bmatrix},\tag{5}$$

where $^j\mathbf{A}_i$ is the 3×3 rotation transformation between the two coordinate systems, and $\mathbf{b}_j$ is the 3×1 position vector from the origin of frame i to the origin of frame j, with components expressed in frame i. The 3×3 matrix, $\tilde{\mathbf{b}}_j$, is an anti-symmetric matrix defined by the rule:

$$\tilde{\mathbf{c}} = \begin{bmatrix} 0 & -c_3 & c_2 \\ c_3 & 0 & -c_1 \\ -c_2 & c_1 & 0 \end{bmatrix}.\tag{6}$$

In spatial notation, inertia matrices are also expressed as 6×6 matrices. An inertia matrix may be defined for each individual link of a chain, as well as the reference member, in its own corresponding coordinate system. For the reference member, this matrix, $\mathbf{I}_0$, is represented as follows:

$$\mathbf{I}_0 = \begin{bmatrix} \bar{\mathbf{I}}_0 & \tilde{\mathbf{h}}_0 \\ \tilde{\mathbf{h}}_0^T & \mathbf{M}_0 \end{bmatrix},\tag{7}$$

where $\mathbf{M}_0$ is a 3×3 diagonal matrix of the mass of the reference member, and $\bar{\mathbf{I}}_0$ is the 3×3 moment of inertia tensor at the origin of coordinate frame 0. The matrix $\bar{\mathbf{I}}_0$ is symmetric and positive definite, but not necessarily diagonal. The 3×3 matrix, $\tilde{\mathbf{h}}_0$, is equal to $m_0 \tilde{\mathbf{s}}_0$, where m_0 is the mass of the reference member, and $\mathbf{s}_0$ is the position vector of the center of gravity of the reference member from the coordinate origin of frame 0. Because $\bar{\mathbf{I}}_0$ and $\mathbf{s}_0$ are defined in coordinate system 0, the matrix $\mathbf{I}_0$ is constant.

To include general joints and contacts with multiple degrees of freedom in a multibody system, an extended model of the interconnections and interactions between individual bodies of that system is required. In this paper, the general joint model of Roberson and Schwertassek [6] is used for this purpose. This model is also used by Brandl, Johanni, and Otter in [5].

Briefly, each interconnection and/or interaction between two bodies in a simple closed-chain mechanism, hereafter referred to as a "general joint", is described in terms of two orthogonal vector spaces, ϕ and ϕ^c. The matrix ϕ is of dimension $6 \times n$, where n represents the number of degrees of freedom of the general joint, and it has full column rank. It represents the free modes of the joint, and its columns make up a basis for this free vector space. We will refer to ϕ as the *motion space* of the general joint. The matrix ϕ^c, which is $6 \times (6 - n)$ and also of full rank, represents the constrained modes of the general joint. It is orthogonal to ϕ, and may be called the *constraint space* of the joint. Both ϕ and ϕ^c are usually resolved in the joint frame, and thus, they are both constant.

III. Dynamic Properties of Individual Chains and Reference Member

Each chain in a simple closed-chain mechanism is governed by the dynamic equations of motion for a single chain. For chain k, $k = 1, \ldots, m$, these are:

$$\tau_k = \mathbf{H}_k\,\ddot{\mathbf{q}}_k + \mathbf{C}_k\,\dot{\mathbf{q}}_k + \mathbf{G}_k + \mathbf{J}_k^T\,\mathbf{f}_k, \tag{8}$$

where

$$
\begin{aligned}
\tau_k &= N_k \times 1 \text{ applied general joint torque/force vector,} \\
\mathbf{q}_k, \dot{\mathbf{q}}_k, \ddot{\mathbf{q}}_k &= N_k \times 1 \text{ general joint position, rate, and acceleration vectors,} \\
\mathbf{H}_k &= N_k \times N_k \text{ joint space inertia matrix,} \\
\mathbf{C}_k &= N_k \times N_k \text{ centripetal/Coriolis matrix,} \\
\mathbf{G}_k &= N_k \times 1 \text{ gravity vector,} \\
\mathbf{J}_k &= 6 \times N_k \text{ Jacobian matrix,}
\end{aligned}
$$

and $\mathbf{f}_k$ is the (6×1) spatial force vector exerted by chain k on the reference member.

Note that $\mathbf{H}_k$, $\mathbf{C}_k$, $\mathbf{G}_k$, and $\mathbf{J}_k$ are functions only of the general joint position and rate vectors, $\mathbf{q}_k$ and $\dot{\mathbf{q}}_k$, respectively. Recall also that the "base" of each chain is the support surface, and the "tip" of each chain touches the reference member. The components of $\mathbf{q}_k$ and τ_k correspond to the general joints of each chain, starting with the joint between link 1 and the support surface and ending with the joint preceding link N. The basic unknowns in Eq. (8) are the general joint accelerations, $\ddot{\mathbf{q}}_k$, and the components of the force vector, $\mathbf{f}_k$, in the constrained directions of the general joint at the tip of chain k.

We may use the dynamic equations of motion to partition the joint acceleration and spatial tip acceleration vectors of each chain into the difference of two terms, one known and one unknown. For each chain, we may write [11]:

$$
\begin{aligned}
\ddot{\mathbf{q}}_k &= (\ddot{\mathbf{q}}_k)_{open} - (\mathbf{H}_k^{-1}\mathbf{J}_k^T)\,\mathbf{f}_k, \tag{9} \\
&= (\ddot{\mathbf{q}}_k)_{open} - \mathbf{\Omega}_k\,\mathbf{f}_k, \tag{10}
\end{aligned}
$$

where $(\ddot{\mathbf{q}}_k)_{open}$ is the vector of joint accelerations for chain k in an open, unconstrained configuration $(\mathbf{f}_k = 0)$, and $\mathbf{\Omega}_k$ is a function of the joint positions for chain k. Likewise, for $\ddot{\mathbf{x}}_k$, the tip acceleration for each chain:

$$
\begin{aligned}
\ddot{\mathbf{x}}_k &= (\ddot{\mathbf{x}}_k)_{open} - (\mathbf{J}_k\,\mathbf{H}_k^{-1}\mathbf{J}_k^T)\,\mathbf{f}_k, \tag{11} \\
&= (\ddot{\mathbf{x}}_k)_{open} - \mathbf{\Lambda}_k^{-1}\mathbf{f}_k, \tag{12}
\end{aligned}
$$

where $(\ddot{\mathbf{x}}_k)_{open}$ is the spatial tip acceleration vector for chain k in an open, unconstrained configuration, and $\mathbf{\Lambda}_k^{-1}$ is the inverse operational space inertia matrix for chain k, defined at the tip of the chain [7,11].

The open-chain terms, $(\ddot{\mathbf{q}}_k)_{open}$ and $(\ddot{\mathbf{x}}_k)_{open}$, are completely defined for each chain given the present state joint positions and rates, $\mathbf{q}_k$ and $\dot{\mathbf{q}}_k$, the applied joint torques/forces in the free directions, τ_k, and the motion of the support surface. An appropriate open-chain Direct Dynamics algorithm may be used to calculate these terms. The $O(N)$ recursive algorithms of [5,10] are very efficient for this computation, and the linear order of computation is highly desirable. Because the joint positions are known, $\mathbf{\Omega}_k$ and $\mathbf{\Lambda}_k^{-1}$ are also defined. Efficient algorithms for $\mathbf{\Omega}_k$ and $\mathbf{\Lambda}_k^{-1}$ for a single chain are derived in [11], including an $O(N)$ recursive algorithm which is the most efficient for $N \geq 6$.

The dynamic behavior of the reference member may be described using a spatial force balance equation for that body. The sum of the spatial forces exerted by each chain on the reference member and any other external spatial forces (including gravity) are equal to the resultant force on the reference member. Using spatial notation, we may write the force balance equation as follows:

$$\mathbf{F}_0 = \sum_{k=1}^{m} {}^0\mathbf{f}_k + \mathbf{g}_0, \tag{13}$$

where

$\mathbf{F}_0$ = 6×1 resultant spatial force vector applied to the reference member,

${}^0\mathbf{f}_k$ = 6×1 spatial force vector applied by chain k to the reference member,

and

$\mathbf{g}_0$ = 6×1 external spatial force vector applied to the reference member (including gravity).

Each force term in Eq. (13) is defined with respect to the coordinate frame attached to the reference member (frame 0). Applying the basic Newton-Euler equations, we may also write the resultant vector, $\mathbf{F}_0$, as follows:

$$\mathbf{F}_0 = \mathbf{I}_0 \mathbf{a}_0 + \mathbf{v}_0 \times \mathbf{I}_0 \mathbf{v}_0, \tag{14}$$
$$= \mathbf{I}_0 \mathbf{a}_0 + \mathbf{b}_0, \tag{15}$$

where

$\mathbf{I}_0$ = 6×6 spatial inertia of the reference member,

$\mathbf{a}_0$ = 6×1 spatial acceleration of the reference member,

$\mathbf{v}_0$ = 6×1 spatial velocity of the reference member.

Both $\mathbf{v}_0$ and $\mathbf{a}_0$ refer to the motion of the coordinate origin of frame 0. The spatial inertia matrix, $\mathbf{I}_0$, is also defined at this point, and it is known and constant. Because $\mathbf{v}_0$ is given for the present state, the velocity-dependent term, $\mathbf{b}_0$, is known. If we combine Eqs. (13) and (15), we finally obtain the following dynamic equation for the reference member:

$$\sum_{k=1}^{m} {}^0\mathbf{f}_k + \mathbf{g}_0 = \mathbf{I}_0 \mathbf{a}_0 + \mathbf{b}_0. \tag{16}$$

In this equation, the basic unknowns are $\mathbf{a}_0$ and the components of ${}^0\mathbf{f}_k$ in the constrained directions of the general joint at the tip of chain k.

IV. Multiple Chain Algorithm

In developing an efficient algorithm for the dynamic simulation of simple closed-chain mechanisms, we are naturally led to consider the relationship between the physical structure of the robotic system and the computational structure of the desired algorithm. Intuitively, it seems apparent that the structural parallelism present in a simple closed-chain mechanism should lead to computational parallelism in the solution of the Direct Dynamics problem for that mechanism.

More specifically, in a simple closed-chain mechanism, the m actuated chains act on the reference member in parallel, and their motion is coupled with that of the reference member. If the reference member is removed, the chains may function independently. Computationally, the physical removal of the reference member corresponds to solving for the forces which are exerted on it by each chain. Once these forces are known, the system is equivalent to a group of independent chains with known tip forces. The joint accelerations may then be computed for each chain separately. Given enough processors (one per chain), the computations for each chain may be carried out in parallel.

We will illustrate the basic methodology of the new simulation algorithm by first examining a simple special case. Consider m manipulators rigidly grasping a common object. Each manipulator has six degrees of freedom, and no chain is in a singular position. For simplicity, we will express all of the relevant equations in absolute coordinates. Because each chain tip is rigidly attached to the reference member, we may write:

$$\ddot{\mathbf{x}}_k = \mathbf{a}_0 \tag{17}$$

for each chain k, $k = 1, \ldots, m$. Thus, the operational space dynamic equation for each chain, as given in Eq. (12), takes the form:

$$\mathbf{a}_0 = (\ddot{\mathbf{x}}_k)_{open} - \Lambda_k^{-1} \mathbf{f}_k. \tag{18}$$

Because no chain is in a singular position, and each chain has a full six degrees of freedom, Λ_k is defined [11]. We may, therefore, solve for the spatial tip force exerted by chain k on the reference member, $\mathbf{f}_k$, as follows:

$$\mathbf{f}_k = \Lambda_k \left[(\ddot{\mathbf{x}}_k)_{open} - \mathbf{a}_0 \right]. \tag{19}$$

With this equation we have established an explicit relationship between the spatial tip force, $\mathbf{f}_k$, and the spatial acceleration of the reference member, $\mathbf{a}_0$. This expression may be used in the reference member dynamic equation, given in Eq. (16), to obtain:

$$\sum_{k=1}^{m} \Lambda_k \left[(\ddot{\mathbf{x}}_k)_{open} - \mathbf{a}_0 \right] = I_0 \, \mathbf{a}_0 + \mathbf{b}_0 - \mathbf{g}_0. \tag{20}$$

The only unknown in Eq (20) is $\mathbf{a}_0$, the spatial acceleration of the reference member. Collecting terms, we may write:

$$\left[I_0 + \sum_{k=1}^{m} \Lambda_k \right] \mathbf{a}_0 = \left[\sum_{k=1}^{m} \Lambda_k \, (\ddot{\mathbf{x}}_k)_{open} - \mathbf{b}_0 + \mathbf{g}_0 \right]. \tag{21}$$

We may now solve for $\mathbf{a}_0$ from this linear system of algebraic equations using any linear system solver. Note that the characteristic matrix is just the sum of the operational space inertia matrices of the individual chains and reference member, and is only 6 × 6. With $\mathbf{a}_0$ known, we may also solve explicitly for the spatial tip force $\mathbf{f}_k$, $k = 1, \ldots, m$, using Eq. (19). Thus, the motion of the reference member and the spatial force exerted at the tip of each chain are completely defined. The simple closed-chain mechanism is effectively decoupled. Each manipulator may now be treated as an independent chain with a known spatial tip force. The joint accelerations for each chain may be computed separately using an appropriate Direct Dynamics algorithm and then integrated to obtain the next state.

The method outlined above is quite straightforward. Of course, the illustrated example represents a special case. We will now develop a similar approach for a general simple closed-chain mechanism. Consider a mechanism with m chains, each with an arbitrary number of degrees of freedom, N. The interaction between each chain tip and the reference member is arbitrary and will be modelled using the general joint model of [6]. To begin, we will derive an explicit relationship between the spatial acceleration of each chain tip and the spatial acceleration of the reference member. The spatial acceleration of the tip of chain k is denoted by $\ddot{\mathbf{x}}_k$. The relative spatial acceleration between the tip of chain k and the reference member, $\ddot{\mathbf{x}}_k^r$, resolved in the orthogonal vector spaces of the general joint between them, may be written:

$$\ddot{\mathbf{x}}_k^r = (\phi)_k \, \alpha_k + (\phi^c)_k \, \alpha_k^c, \tag{22}$$

C-2

where $(\phi)_k$ and $(\phi^c)_k$ are the motion space and constraint space of the general joint at the tip of chain k, respectively. The quantities α_k and α_k^c are the corresponding components of relative acceleration in the free and constrained directions. For each chain, $(\phi)_k$, $(\phi^c)_k$, and α_k^c are known, while α_k is unknown. The sum of $\ddot{\mathbf{x}}_k$ and $\ddot{\mathbf{x}}_k^r$ is just the spatial acceleration of the reference member on the far side of the general joint between it and chain k, $\mathbf{a}_0^k$. Thus, we may write:

$$\mathbf{a}_0^k = \ddot{\mathbf{x}}_k + \ddot{\mathbf{x}}_k^r, \tag{23}$$

$$= \ddot{\mathbf{x}}_k + (\phi)_k \, \alpha_k + (\phi^c)_k \, \alpha_k^c. \tag{24}$$

We may also express $\mathbf{a}_0^k$ in terms of the spatial acceleration of the reference member, $\mathbf{a}_0$, as follows:

$$\mathbf{a}_0^k = \mathbf{X}_0^k \, \mathbf{a}_0 + \zeta_0^k, \tag{25}$$

where $\mathbf{X}_0^k \equiv {}^N\mathbf{X}_0^k$ is the spatial transformation between coordinate frame 0 and the coordinate frame associated with the general joint at the tip of chain k. The quantity ζ_0^k is the 6×1 bias acceleration vector which is a function of the position and spatial velocity of the reference member. Because the present state of the entire system is given, both $\mathbf{X}_0^k$ and ζ_0^k are known.

Equating the two expressions above for $\mathbf{a}_0^k$, we obtain the following:

$$\ddot{\mathbf{x}}_k + (\phi)_k \, \alpha_k + (\phi^c)_k \, \alpha_k^c = \mathbf{X}_0^k \, \mathbf{a}_0 + \zeta_0^k. \tag{26}$$

This equation matches the spatial accelerations at the coupling point between chain k and the reference member, giving an explicit relationship between $\ddot{\mathbf{x}}_k$ and $\mathbf{a}_0$ when the coupling is arbitrary. The basic unknowns in Eq. (26) are $\ddot{\mathbf{x}}_k$, α_k, and $\mathbf{a}_0$. All other vectors and matrices may be computed rather simply from the initial information given for the simulation problem.

To decouple the chains and the reference member, we need an explicit mathematical relationship between the spatial force exerted by chain k on the reference member, $\mathbf{f}_k$, and the spatial acceleration of the reference member, $\mathbf{a}_0$. Equation (26) relates the spatial acceleration of the reference member and the spatial acceleration of the tip of chain k. We may eliminate α_k, the unknown components of the relative acceleration, by projecting Eq. (26) onto the constraint space of the corresponding general joint as follows:

$$(\phi^c)_k^T \left[\ddot{\mathbf{x}}_k + (\phi)_k \, \alpha_k + (\phi^c)_k \, \alpha_k^c \right] = (\phi^c)_k^T \left[\mathbf{X}_0^k \, \mathbf{a}_0 + \zeta_0^k \right]. \tag{27}$$

By definition [11]:

$$(\phi^c)_k^T \, (\phi)_k = \mathbf{0}, \tag{28}$$

and .

$$(\phi^c)_k^T \, (\phi^c)_k = \mathbf{1}. \tag{29}$$

Thus, we may write:

$$(\phi^c)_k^T \, \ddot{\mathbf{x}}_k + \alpha_k^c = (\phi^c)_k^T \left[\mathbf{X}_0^k \, \mathbf{a}_0 + \zeta_0^k \right]. \tag{30}$$

Equation (12) defines $\ddot{\mathbf{x}}_k$ in terms of the desired force vector, $\mathbf{f}_k$. If we combine Eqs. (12) and (30), we obtain:

$$(\phi^c)_k^T \left[(\ddot{\mathbf{x}}_k)_{open} - \Lambda_k^{-1}\mathbf{f}_k \right] = (\phi^c)_k^T \left[\mathbf{X}_0^k \, \mathbf{a}_0 + \zeta_0^k \right] - \alpha_k^c, \tag{31}$$

or

$$\left[(\phi^c)_k^T\,\Lambda_k^{-1}\right]\,\mathbf{f}_k = \left[\alpha_k^c - (\phi^c)_k^T\,\zeta_0^k + (\phi^c)_k^T\,(\ddot{\mathbf{x}}_k)_{open}\right] - \left[(\phi^c)_k^T\,\mathbf{X}_0^k\right]\,\mathbf{a}_0. \tag{32}$$

The first bracketed term on the right side of Eq. (32) is completely known. The only unknowns in this equation are the constraint components of the force vector, $\mathbf{f}_k$, and the spatial acceleration, $\mathbf{a}_0$. We may now pursue an explicit relationship between these two vectors.

Like the relative acceleration vector, $\mathbf{f}_k$ may also be resolved in the orthogonal vector spaces of the general joint at the tip of chain k as follows:

$$\mathbf{f}_k = (\phi)_k\,\mathbf{h}_k + (\phi^c)_k\,\mathbf{h}_k^c, \tag{33}$$

where $\mathbf{h}_k$ is the vector of known force components in the free directions, and $\mathbf{h}_k^c$ is the vector of unknown force components in the constrained directions. Combining Eqs. (32) and (33), we obtain:

$$\begin{aligned}
(\phi^c)_k^T\,\Lambda_k^{-1}\,[(\phi)_k\,\mathbf{h}_k + (\phi^c)_k\,\mathbf{h}_k^c] &= \left[\alpha_k^c - (\phi^c)_k^T\,\zeta_0^k + (\phi^c)_k^T\,(\ddot{\mathbf{x}}_k)_{open}\right] \\
&\quad - \left[(\phi^c)_k^T\,\mathbf{X}_0^k\right]\,\mathbf{a}_0.
\end{aligned} \tag{34}$$

If the spatial acceleration of the reference member is known, we may find an explicit solution for the unknown force components at the tip of chain k from the following set of linear algebraic equations:

$$\begin{aligned}
\left[(\phi^c)_k^T\,\Lambda_k^{-1}(\phi^c)_k\right]\,\mathbf{h}_k^c &= \left\{\alpha_k^c - (\phi^c)_k^T\left[\zeta_0^k - (\ddot{\mathbf{x}}_k)_{open} + \Lambda_k^{-1}(\phi)_k\,\mathbf{h}_k\right]\right\} \\
&\quad - \left[(\phi^c)_k^T\,\mathbf{X}_0^k\right]\,\mathbf{a}_0, \tag{35} \\
&= \mathbf{S}_k - \left[(\phi^c)_k^T\,\mathbf{X}_0^k\right]\,\mathbf{a}_0, \tag{36}
\end{aligned}$$

where $\mathbf{S}_k$ is known. Even when $\mathbf{a}_0$ is unknown, we may still find a solution for $\mathbf{h}_k^c$ in terms of the unknown $\mathbf{a}_0$. The solution will have the following form:

$$\begin{aligned}
\mathbf{h}_k^c &= \mathbf{M}_k\left[\mathbf{S}_k - (\phi^c)_k^T\,\mathbf{X}_0^k\,\mathbf{a}_0\right], \tag{37} \\
&= \mathbf{M}_k\,\mathbf{S}_k - \left[\mathbf{M}_k\,(\phi^c)_k^T\,\mathbf{X}_0^k\right]\,\mathbf{a}_0. \tag{38}
\end{aligned}$$

If $(n_c)_k$ is the number of degrees of constraint for the general joint at the tip of chain k, then $\mathbf{M}_k$ is the $(n_c)_k \times (n_c)_k$ transformation matrix which solves for $\mathbf{h}_k^c$. By carefully considering the rank of the coefficient matrix, this general solution can still be used for a chain in a singular position or a chain with less than six original degrees of freedom [11]. This solution procedure requires $O[(n_c)_k^3]$ scalar operations.

Note that if chain k is rigidly grasping the reference member, then the constraint space for this general joint, $(\phi^c)_k$, is the 6×6 identity matrix. In this case, also note that α_k^c and $\mathbf{h}_k$ are identically zero for each chain. If chain k has six degrees of freedom and is not in a singular position, then $\mathbf{M}_k$ will be exactly equal to Λ_k, the operational space inertia matrix for chain k, and the solution for $\mathbf{h}_k^c$ will be:

$$\mathbf{h}_k^c = \Lambda_k\left[\mathbf{S}_k - \mathbf{X}_0^k\,\mathbf{a}_0\right]. \tag{39}$$

This solution corresponds to the simple example discussed at the beginning of this section, but now expressed in local coordinates.

Given the general solution for h_k^c in Eq. (38), the force vector, f_k, may now be written:

$$f_k = (\phi)_k h_k + (\phi^c)_k h_k^c, \tag{40}$$

$$= [(\phi)_k h_k + (\phi^c)_k M_k S_k] - \left[(\phi^c)_k M_k (\phi^c)_k^T X_0^k\right] a_0, \tag{41}$$

$$= P_k - R_k a_0, \tag{42}$$

where P_k and R_k are of dimension 6×1 and 6×6, respectively, and both may be computed from known quantities. We now have an explicit equation relating the force vector exerted by chain k and the spatial acceleration of the reference member. We may combine this information with the dynamic equation for the reference member to solve for a_0 explicitly.

The dynamic equation for the reference member given in Eq. (16) may be rewritten as follows:

$$\sum_{k=1}^{m} (X_0^k)^T f_k + g_0 = I_0 a_0 + b_0. \tag{43}$$

where f_k is the spatial force exerted by chain k on the reference member, expressed in the coordinate frame of the general joint at the chain tip. If the expression for f_k in Eq. (42) is used in Eq. (43), we obtain:

$$\sum_{k=1}^{m} (X_0^k)^T (P_k - R_k a_0) = I_0 a_0 + b_0 - g_0. \tag{44}$$

Summing like terms, we may write:

$$\left[I_0 + \sum_{k=1}^{m} (X_0^k)^T R_k\right] a_0 = \left[\sum_{k=1}^{m} (X_0^k)^T P_k - b_0 + g_0\right], \tag{45}$$

or, expanding R_k,

$$\left[I_0 + \sum_{k=1}^{m} (X_0^k)^T (\phi^c)_k M_k (\phi^c)_k^T (X_0^k)\right] a_0 = \left[\sum_{k=1}^{m} (X_0^k)^T P_k - b_0 + g_0\right]. \tag{46}$$

In Eq. (46), the 6×1 spatial acceleration vector of the reference member, a_0, is the only unknown. We may find a solution for a_0 from the given set of linear algebraic equations using any efficient linear system solver. Because the required system solution always involves a 6×6 coefficient matrix, the computational cost of solving for a_0 is constant.

The coefficient matrix of a_0 in Eq. (46) represents the combined inertial properties of all the chains and the reference member. The inertial properties of each chain are first projected to the tip of that chain by computing the inverse operational space inertia matrix, Λ_k^{-1}. Along the free directions of the general joint which connects the chain tip and the reference member, the projected inertia of the chain is not felt by the reference member. Along the constrained directions of the joint, however, the corresponding components of Λ_k^{-1} are reflected across to the reference member. These components, spatially transformed to the coordinate origin of frame 0, are combined with the spatial inertia of the reference member, I_0. This combination represents the effective operational space inertia of the simple closed-chain mechanism defined at the coordinate origin of frame 0. It is the effective inertia "felt" by the reference member in the present state. The bracketed term on the right side of Eq. (46) represents the spatial forces which act on the reference member at the given instant.

Once the spatial acceleration of the reference member is known, the spatial force vector applied by chain k to the reference member is defined by Eq. (42). That is, with a_0 given, we may compute f_k as follows:

$$\mathbf{f}_k = \mathbf{P}_k - \mathbf{R}_k \, \mathbf{a}_0. \tag{47}$$

Recall that $\mathbf{f}_k$ is defined with respect to the coordinate frame of the general joint between chain k and the reference member. The explicit knowledge of $\mathbf{f}_k$ allows us to treat chain k as an independent chain with a known tip force. We may now solve for the general closed-chain joint accelerations for chain k using Eq. (10), repeated here for convenience:

$$\ddot{\mathbf{q}}_k = (\ddot{\mathbf{q}}_k)_{open} - \mathbf{\Omega}_k \, \mathbf{f}_k. \tag{48}$$

The application of Eq. (48) to every actuated chain in the simple closed-chain mechanism results in a complete solution to the Direct Dynamics problem for this robotic system. The next state positions and velocities may be computed by integrating the appropriate quantities for each chain and the reference member. As discussed in [11], small amounts of negative position and rate feedback may be employed to counteract the drift which is a result of the integration process, and which would violate the kinematic constraints.

The algorithm developed here for simple closed-chain mechanisms may be presented as a series of five steps. They are as follows:

1. The Open Chain Solution,

2. Calculation of the Spatial Acceleration of the Reference Member,

3. Calculation of the Spatial Chain Tip Forces,

4. Calculation of the Closed-Chain Joint Accelerations,

5. Integration for the Next State.

The fundamental computations required in each of these steps are summarized in Table 1. In Step 1, the Direct Dynamics problem is solved for each chain of the mechanism assuming that the reference member has been removed and each chain is in an open, unconstrained state. The general open-chain acceleration vectors, $(\ddot{\mathbf{q}}_k)_{open}$ and $(\ddot{\mathbf{x}}_k)_{open}$, are computed for each chain, along with the position-dependent matrices, $\mathbf{\Omega}_k$ and $\mathbf{\Lambda}_k^{-1}$. In Step 2, Eq. (45) is used to find an explicit solution for $\mathbf{a}_0$, the spatial acceleration of the reference member, via linear system solution. The quantities $(\mathbf{M}_k \, \mathbf{S}_k)$ and $[\mathbf{M}_k \, (\phi^c)_k^T \, \mathbf{X}_0^k]$, required for both $\mathbf{P}_k$ and $\mathbf{R}_k$, are computed in the determination of the explicit relationship between $\mathbf{h}_k^c$ and $\mathbf{a}_0$. This relationship is found by linear system solution using Eq. (36), with the solution taking the form of Eq. (38). In Step 3, this solution is used in Eq. (47) to solve for the spatial force vector exerted on the reference member by each chain. In Step 4, the general closed-chain joint accelerations are computed for each chain using Eq. (48), given the spatial tip force vector. In Step 5, the appropriate rates and accelerations are integrated to obtain the next state positions and rates for the system.

Note that the first step may be carried out for all chains in parallel, if enough processors are available (one per chain). Once the second step is complete and $\mathbf{a}_0$ is known, the third, fourth, and fifth steps may also be carried out for all chains simultaneously. Thus, taking advantage of the structural parallelism inherent in the simple closed-chain system has led to parallelism in the computational structure of the simulation algorithm.

V. Computational Requirements

We will now consider the computational requirements of the dynamic simulation algorithm for simple closed-chain mechanisms. First, the number of scalar operations required for each chain of the mechanism will be tabulated, followed by the number of operations required to compute the spatial

Table 1: Dynamic Simulation Algorithm for Simple Closed-Chain Mechanisms

Given: α_k^c, $\mathbf{h}_k$, $(\phi^c)_k$, $(\phi)_k$, and with

$\mathbf{X}_0^k$, $\mathbf{b}_0$, $\mathbf{g}_0$, ζ_0^k determined from the reference member state;

Step 1.   Compute $(\ddot{\mathbf{q}}_k)_{open}$, $(\ddot{\mathbf{x}}_k)_{open}$, Ω_k, and Λ_k^{-1}; $k = 1, \ldots, m$.

Step 2.   Solve for $\mathbf{a}_0$:

$$\left[\mathbf{I}_0 + \sum_{k=1}^m (\mathbf{X}_0^k)^T \mathbf{R}_k \right] \mathbf{a}_0 = \left[\sum_{k=1}^m (\mathbf{X}_0^k)^T \mathbf{P}_k - \mathbf{b}_0 + \mathbf{g}_0 \right],$$

with

$$\begin{aligned}
\mathbf{P}_k &= \left[(\phi)_k \, \mathbf{h}_k + (\phi^c)_k \, \mathbf{M}_k \, \mathbf{S}_k \right], \\
\mathbf{R}_k &= \left[(\phi^c)_k \, \mathbf{M}_k \, (\phi^c)_k^T \, \mathbf{X}_0^k \right], \\
\mathbf{S}_k &= \alpha_k^c - (\phi^c)_k^T \left[\zeta_0^k - (\ddot{\mathbf{x}}_k)_{open} + \Lambda_k^{-1} \, (\phi)_k \, \mathbf{h}_k \right],
\end{aligned}$$

and where $(\mathbf{M}_k \, \mathbf{S}_k)$ and $\left[\mathbf{M}_k \, (\phi^c)_k^T \, \mathbf{X}_0^k \right]$ are determined by the solution of:

$$(\mathbf{M}_k^{-1}) \, \mathbf{h}_k^c = \left[(\phi^c)_k^T \, \Lambda_k^{-1} \, (\phi^c)_k \right] \, \mathbf{h}_k^c = \mathbf{S}_k - \left[(\phi^c)_k^T \, \mathbf{X}_0^k \right] \, \mathbf{a}_0.$$

Step 3.   Solve for $\mathbf{f}_k$; $k = 1, \ldots, m$:

$$\mathbf{f}_k = \mathbf{P}_k - \mathbf{R}_k \, \mathbf{a}_0.$$

Step 4.   Solve for $\ddot{\mathbf{q}}_k$; $k = 1, \ldots, m$:

$$\ddot{\mathbf{q}}_k = (\ddot{\mathbf{q}}_k)_{open} - \Omega_k \, \mathbf{f}_k.$$

Step 5.   Integrate to obtain the next state positions and rates for the system.

Table 2: Computations Per Chain in the Simple Closed-Chain Dynamic Simulation Algorithm

Calculation	#Mult.	#Add.	#Mult. $(N = 6, n_c = 3)$	#Add. $(N = 6, n_c = 3)$
$\ddot{q}_{open}, \ddot{x}_{open}$	$250N - 182$	$220N - 167$	1318	1153
Ω, Λ^{-1}	$400N - 621$	$320N - 528$	1779	1392
P, R	$\frac{1}{6}n_c^3 + 6\frac{1}{2}n_c^2 + 5\frac{1}{3}n_c + 26$	$\frac{1}{6}n_c^3 + 6n_c^2 + \frac{5}{6}n_c + 10$	105	71
$X^T P, X^T R$	$36n_c + 20$	$36n_c - 24$	128	84
f	36	36	36	36
$\ddot{q}$	$6N$	$6N$	36	36
Total:	$656N - 767$ $+(\frac{1}{6}n_c^3 + 6\frac{1}{2}n_c^2 + 41\frac{1}{3}n_c + 46)$	$546N - 659$ $+(\frac{1}{6}n_c^3 + 6n_c^2 + 36\frac{5}{6}n_c - 14)$	3402	2772

acceleration of the reference member. The computational complexity of the complete algorithm will then be discussed, and the parallel implementation of this algorithm will be considered.

Table 2 lists the number of scalar operations (multiplications, additions) required in the simulation algorithm for each chain of a simple closed-chain mechanism. The operations are tabulated for the case of an N degree-of-freedom serial-link chain with simple revolute and/or prismatic joints only. The $O(N)$ Direct Dynamics algorithm of [5] is used to compute the open-chain terms, $\ddot{q}_{open}$ and $\ddot{x}_{open}$. The $O(N)$ Force Propagation Method of [11] is used to compute Ω and Λ^{-1}. All $\ddot{q}_{open}, \ddot{x}_{open}$, Ω, and Λ^{-1} are computed in Step 1 of the simulation algorithm.

In Step 2 of the simulation algorithm, the spatial acceleration of the reference member is calculated using Eq. (45). For this task, $X^T P$ and $X^T R$ must be computed for each chain. The number of operations required to compute P, R, $X^T P$, and $X^T R$ are also listed in Table 2. In this case, the number of operations is a function of the number of degrees of constraint at the general joint between the chain tip and the reference member (n_c). This number can never be greater than six. The computational complexity of these calculations is $O(n_c^3)$ due to the linear system solution required in the computation of both P and R (see Table 1).

The spatial force vector, f, exerted by each chain on the reference member, and the closed-chain joint accelerations for the chain, $\ddot{q}$, are calculated in Steps 3 and 4 of the simulation algorithm, respectively. The appropriate equations are given in Table 1. The operations required to calculate these vectors complete the table. The operations required for the special case of $N = 6$ and $n_c = 3$ are given in the last two columns of Table 2. This value of n_c could correspond to a hard point contact between a manipulator tip and surface of a load object when the tip is not slipping.

Given the computations required for each individual chain, the number of scalar operations needed to compute the spatial acceleration of the reference member, a_0, is given in Table 3. Equation (45) is used to obtain the solution, which requires $O(m)$ spatial additions and a single 6×6 symmetric linear system solution. Thus, the number of operations required for a_0 is a function only of m, the number of chains in the simple closed-chain mechanism. The example of three chains $(m = 3)$ is given in the last two columns of this table.

To determine the total number of scalar operations required to simulate the entire simple closed-chain mechanism, the number of operations required for a single chain is simply multiplied by m,

85

Table 3: Computations for the Spatial Acceleration of the Reference Member

Calculation	#Mult.	#Add.	#Mult. $(m = 3)$	#Add. $(m = 3)$
$\mathbf{a_0}$	86	$27m + 71$	86	152

the number of chains, and added to the computations required for $\mathbf{a_0}$ for the same number of chains. Thus, the computational complexity of the complete simulation algorithm is $O(mN)$ for a given value of $n_c \leq 6$.

The total computational complexity discussed in the previous section only considered the execution of the simulation algorithm on a single processor. In order to speed up the simulation, parallel processing may be investigated. If a single processor is used for the entire system of m chains, the computational complexity of the simulation algorithm is $O(mN)$ for a given $n_c \leq 6$. Given $\mathbf{a_0}$, all computations for each chain may be carried out independently. Thus, if m processors are available, the computational tasks associated with each chain may be performed in parallel, and the computational complexity of the operations required for the m chains may be reduced to $O(N)$. Of course, the computations required to compute $\mathbf{a_0}$ must also be considered. These operations may also be implemented in parallel on the m available processors. Equation (45) requires $O(m)$ spatial additions to compute $\mathbf{a_0}$. On $(m + 1)/2$ parallel processors, this task may be carried out in $O(\log_2 m + 1)$ operations by using the recursive doubling approach [12]. Thus, on m parallel processors, the computational complexity of the entire dynamic simulation algorithm may be reduced to $O(N) + O(\log_2 m + 1)$.

VI. Summary and Conclusions

In this paper, a general and efficient dynamic simulation algorithm for simple closed-chain mechanisms was derived. The algorithm is applicable to both Type 0 and Type 1 mechanisms. Both types of mechanisms are modelled in a convenient and general manner through the use of the general joint concept. The operational space inertia matrix of each chain is used to project the dynamic properties of the chain to its tip where it is coupled to the reference member. By combining the operational space inertia of each chain with the model of the general joint at each chain tip, a solution may be found for the spatial acceleration of the reference member and the spatial force vector exerted on it by each chain. Once the force vectors are completely defined, the system is effectively decoupled, and the joint accelerations for each chain may be computed separately.

The computational complexity of the new simulation algorithm is $O(mN)$ when implemented on a single processor. The linear dependence on N is a significant improvement over previous simulation algorithms such as that presented in [1]. The computational complexity of the new algorithm may be further reduced to $O(N) + O(\log_2 m + 1)$ if it is implemented on m processors in parallel.

VII. Acknowledgments

References

[1] S. Y. Oh and D. E. Orin, "Dynamic Computer Simulation of Multiple Closed-Chain Robotic Mechanisms," in *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, pp. 15–20, San Francisco, CA, April 1986.

[2] R. H. Lathrop, "Constrained (Closed-Loop) Robot Simulation by Local Constraint Propagation," in *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, pp. 689–694, San Francisco, CA, April 1986.

[3] H. Brandl, R. Johanni, and M. Otter, "An Algorithm for the Simulation of Multibody Systems with Kinematic Loops," in *Proceedings of the IFToMM Seventh World Congress on the Theory of Machines and Mechanisms*, Sevilla, Spain, September 1987.

[4] G. Rodriguez and K. Kreutz, "Recursive Mass Matrix Factorization and Inversion: An Operator Approach to Open- and Closed-Chain Multibody Dynamics," Technical Report 88-11, Jet Propulsion Laboratory, Pasadena, CA, March 1988.

[5] H. Brandl, R. Johanni, and M. Otter, "A Very Efficient Algorithm for the Simulation of Robots and Similar Multibody Systems Without Inversion of the Mass Matrix," in *Proceedings of IFAC/IFIP/IMACS International Symposium on the Theory of Robots*, Vienna, Austria, December 1986.

[6] R. E. Roberson and R. F. Schwertassek, *Introduction to the Dynamics of Multibody Systems*. Springer-Verlag, Berlin, 1987.

[7] O. Khatib, "A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 1, pp. 43–53, 1987.

[8] K. W. Lilly and D. E. Orin, "$O(N)$ Recursive Algorithm for the Operational Space Inertia Matrix of a Robot Manipulator," submitted to *11th IFAC World Congress*, August 1990.

[9] R. Featherstone, *Robot Dynamics Algorithms*. Boston: Kluwer Academic Publishers, 1987.

[10] R. Featherstone, "The Calculation of Robot Dynamics Using Articulated-Body Inertias," *The International Journal of Robotics Research*, vol. 2, no. 1, pp. 13–30, Spring 1983.

[11] K. W. Lilly, "Efficient Dynamic Simulation of Multiple Chain Robotic Mechanisms," Ph.D. Thesis, The Ohio State University, 1989.

[12] M. Amin-Javaheri and D. E. Orin, "Parallel Algorithms for Computation of the Manipulator Inertia Matrix," in *Proceedings of NASA Conference on Space Telerobotics*, Pasadena, CA, Jan./Feb. 1989.