Multibody System Dynamics **9:** 185–212, 2003.

Improved 'Order-N' Performance Algorithm for the Simulation of Constrained Multi-Rigid-Body Dynamic Systems

K.S. ANDERSON and J.H. CRITCHLEY
Department of Mechanical, Aeronautical and Nuclear Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, U.S.A.; E-mail: {anderk5,critch}@rpi.edu

(Received: 20 August 2001; accepted in revised form: 18 October 2002)

Abstract. This paper presents an algorithm for the efficient numerical analysis and simulation of modest to heavily constrained multi-rigid-body dynamic systems. The algorithm can accommodate the spatial motion of general multi-rigid-body systems containing arbitrarily many closed loops in $O(n+m)$ operations overall for systems containing n generalized coordinates, and m independent algebraic constraints. The presented approach does not suffer from the performance (speed) penalty encountered by most other of the so-called '$O(n)$' state-space formulations, when dealing with constraints which tend to actually show $O(n+m+nm+nm^2+m^3)$ performance. Additionally, these latter formulations may require additional constraint violation stabilization procedures (e.g. Baumgarte's method, coordinate partitioning, etc.) which can contribute significant additional computation. The presented method suffers less from this difficulty because the loop closure constraints at both the velocity and acceleration level are directly embedded within the formulation. Due to these characteristics, the presented algorithm offers superior computing performance relative to other methods in situations involving both large n and m.

Key words: recursive algorithm, constrained system, coordinate reduction.

Nomenclature

$\underline{a}^k$ = matrix representation of acceleration of center of mass k^* in the Newtonian reference frame N

$\underline{a}_t^k$ = acceleration remainder term associated with body k in N; This is all terms of $\underline{a}^k$ which are not explicit in $\dot{u}$'s

$\underline{\mathcal{A}}^k$ = the generalized acceleration matrix of body k in N

$\underline{\bar{\mathcal{A}}}^k$ = that portion of the generalized acceleration matrix of body k in N which is explicit in the unknown state derivatives $\underline{\dot{u}}$

$\underline{\mathcal{A}}_t^k$ = that portion of the generalized acceleration matrix of body k in N which is not explicit in the unknown state derivatives $\underline{\dot{u}}$

${}^{0_i}\underline{\mathcal{A}}^k$ = the generalized acceleration matrix of k_i in reference frame 0_i, which are associated with closed loop i

${}^{0_i}\underline{\bar{\mathcal{A}}}^k$ = that portion of the generalized acceleration matrix of k_i in 0_i which is explicit in the unknown state derivatives $\underline{\dot{u}}$

${}^{0_i}\underline{\mathcal{A}}_t^k$ = that portion of the generalized acceleration matrix of k_i in 0_i which is not explicit in

the unknown state derivatives $\underline{\dot{u}}$

$\underline{C}$ = invertible transformation matrix relating $\underline{\dot{q}}$ to $\underline{u}$

$\underline{\mathcal{C}}_k$ = direction cosine matrix relating the basis vectors fixed in body k to those in proximal body Pr[k]

Dist[k] = distal body set associated with body k

$\underline{D}$ = matrix used in relating $\underline{\dot{q}}$ to $\underline{u}$, and commonly associated with prescribed motions

$\underline{E}_i$ = expansion matrix which converts m_i constraint load measure numbers to constraint load matrix $\underline{\mathcal{F}}_{c_i}$

$\underline{\mathcal{F}}^k$ = recursive generalized force matrix for body k

$\underline{\widehat{\mathcal{F}}}^k$ = articulated body force associated with body k

$\underline{{}^{0_i}\hat{\mathcal{F}}}^k$ = articulated body force associated with body k for $0_i < k_i \leq p_i$

$\underline{f}_{c_i}$ = matrix representation of constraint force used to close i-th closed loop

$\underline{\mathcal{F}}_{c_i}$ = constraint load matrix i-th closed loop

$\underline{I}^{k/k^*}$ = central inertia matrix of body k

$\underline{\mathcal{I}}^k$ = generalized inertia matrix of body k

$\underline{\hat{\mathcal{I}}}^k$ = *articulated body inertia* matrix of body k, associated with acceleration $\underline{\bar{\mathcal{A}}}^k$

$\underline{{}^{0_i}\hat{\mathcal{I}}}^k$ = *articulated body inertia* matrix of body k, associated with acceleration $\underline{{}^{0_i}\bar{\mathcal{A}}}^k$

$\underline{\hat{\mathcal{I}}}^{k;0_i}$ = *articulated body inertia* matrix of body k, associated with acceleration $\underline{\bar{\mathcal{A}}}^{0_i}$

k = index representing an arbitrary system body k (global numbering)

k_i = index representing an arbitrary body k within closed loop i (local loop numbering)

k^* = center of mass of body k

$\underline{K}$ = right-hand side of system equations of motion, representing applied forces, as well as centripetal and Coriolis portions of inertia forces

m = total number of independent system constraints

m_i = total number of constraints associated with closed loop i

$\underline{M}$ = system mass matrix

$\underline{\mathcal{M}}_k$ = matrix of terms associated with generalized speed $\underline{u}_k$ which would be found on the diagonal of a partially triangularized (decomposed) system mass matrix

n = total number of system generalized coordinates

n_i = total number of generalized coordinates associated with i-th closed loop constraint equations

n_L = total number of closed loops

N = Newtonian reference frame

0_i = base body (primary reference frame) of closed loop i

p_i = body containing the highest independent degree-of-freedom (local loop numbering) within closed loop i

$\underline{\mathcal{P}}_k^r$ = partial velocity matrix for body k associated with generalized speeds $\underline{u}_r$

$\underline{\widetilde{\mathcal{P}}}_r^k$ = nonholonomic partial velocity matrix for body k associated with independent generalized speeds $\underline{u}_r$

Pr[k] = proximal (parent) body set associated with body k

$\underline{q}$ = the $n \times 1$ matrix of generalized coordinates used to describe the configuration of the system

$\underline{\dot{q}}$ = the first time derivative of the system generalized coordinates

$\underline{R}^k$ = matrix representation of resultant of all nonconstraint forces acting on body k

$\underline{\mathcal{S}}^k$ = basis consistent *shift matrix* which converts system of forces acting through mass center of k to equivalent system acting through point of k instantaneously coincident with the mass center of its proximal body Pr[k]

$\underline{{}^j\mathcal{S}}^k$ = *shift* matrix which transforms a force system acting through mass center of k to an

equivalent system acting through point of k, instantaneously coincident with the mass center of body j

t = time

$\underline{t}_{c_i}$ = matrix representation of constraint torque used do close i-th closed loop

$\underline{T}^k$ = matrix representation of all moments acting on body k

$\underline{\mathcal{T}}^k$ = local *triangularization matrix* associated with body k

$\underline{\widehat{\mathcal{T}}}_i^k$ = constraint load *triangularization matrix* associated with body k and closed loop i

$\underline{u}$ = the system *generalized speeds* which characterize the motion of the system

$\underline{u}_k$ = system generalized speeds which are directly associated with the motion of body k relative to its parent body

$\underline{\dot{u}}$ = system *generalized accelerations* to be determined and temporally integrated

$\underline{\dot{u}}_k$ = system generalized accelerations which are directly associated with the motion of body k relative to its parent body

$\underline{U}$ = appropriately dimensioned identity matrix

$\mathbf{v}^{k^*}$ = velocity of the center of mass of body k in reference frame N

$\mathbf{v}_r^{k^*}$ = partial velocity of the center of mass of body k in reference frame N associated with u_r

$\mathbf{v}_t^{k^*}$ = *velocity remainder term* associated with the center of mass of body k in reference frame N

$\underline{\mathcal{V}}^k$ = the generalized velocity matrix, which relates the velocity of body k to reference frame N

$\underline{\bar{\mathcal{V}}}^k$ = portion of the body k generalized velocity matrix which is explicit in the generalized speeds $\underline{u}$

$\underline{\mathcal{V}}_t^k$ = body k generalized velocity remainder term matrix, which relates the velocity of body k to reference frame N

${}^{0_i}\underline{\mathcal{V}}^k$ = the generalized velocity matrix, which relates the velocity of k_i mass center to that of loop base body 0_i

${}^{0_i}\underline{\bar{\mathcal{V}}}^k$ = portion of the generalized velocity matrix ${}^{0_i}\underline{\mathcal{V}}^k$, which is explicit in the generalized speeds $\underline{u}$

${}^{0_i}\underline{\mathcal{V}}_t^k$ = that portion of the generalized velocity matrix ${}^{0_i}\underline{\mathcal{V}}^k$, which is not explicit in the generalized speeds $\underline{u}$

α^k = angular acceleration of body k in Newtonian reference frame N

α_t^k = *angular acceleration remainder terms* of body k; this represents all terms of α^k which are not explicit in $\dot{u}$'s

$\underline{\delta}^j$ = useful intermediate quantity associated with recursive treatment of dependent generalized speeds and associated state derivatives of the closed loop under consideration

$\underline{\Delta}^{k_i}$ = useful intermediate quantity associated with recursive treatment of body k_i of the closed loop under consideration

$\boldsymbol{\gamma}^k$ = position vector from body Pr[k] mass center to body k mass center

$\underline{\Gamma}^k$ = useful intermediate quantity associated with recursive treatment of dependent degrees of freedom of body k within the closed loop under consideration

$\underline{\zeta}_i^k$ = coefficient matrix to unknown constraint load measure numbers $\underline{\lambda}_i$ associated with loop i within expression for constrained system generalized acceleration

$\underline{\bar{\zeta}}_i^k$ = intermediate quantity useful in the recursive determination of $\underline{\zeta}_i^k$

$\underline{\eta}^k$ = portion of constrained system generalized accelerations $\underline{\dot{u}}$ which is not explicit in constraint load measure numbers $\underline{\lambda}$

$\underline{\bar{\eta}}^k$ = intermediate quantity useful in the recursive determination of $\underline{\eta}^k$

$\underline{\Theta}^k$ = intermediate quantity appearing in the solution for coupled loop systems
$\underline{\lambda}$ = Lagrange multipliers
$\underline{\tau}^{k_i}$ = useful intermediate quantity associated with recursive treatment of body k_i of the closed loop under consideration
$\underline{\Xi}^k$ = useful intermediate quantity associated with recursive treatment of dependent degrees of freedom of body k within the closed loop under consideration
$\underline{\Phi}$ = set of m system algebraic constraint equations
$\underline{\Phi}_{,q}$ = system constraint Jacobian
$\underline{\chi}^{p+j}$ = useful intermediate quantity associated with recursive treatment of dependent generalized speeds and associated state derivatives $p + j$ of the closed loop under consideration
$\underline{\Psi}$ = portion of velocity level constraint equations which is not explicit in generalized speeds $\underline{u}$; this term is usually associated with specified motions
$\underline{\psi}$ = portion of acceleration level constraint equations which is not explicit in generalized speeds time derivatives $\underline{\dot{u}}$
$\boldsymbol{\omega}^k$ = angular velocity of body k in reference frame N
$\boldsymbol{\omega}_r^k$ = partial angular velocity of body k in reference frame N associated with u_r
$\boldsymbol{\omega}_t^k$ = *angular velocity remainder term* associated with body k in reference frame N
$\underline{\omega}_\times^k$ = matrix equivalent to vector cross-product $\boldsymbol{\omega}^k\times$

1. Introduction

Computational efficiency of multibody system simulations has been receiving increasing attention since the first $O(n)$ (computational effort per temporal integration step increases as linear function of the number of generalized coordinates) algorithm developed by Vereshchagin in 1975 [30]. This work marked a significant departure in potential algorithm computational cost from that possible with more traditional formulations, which tended to offer $O(n^3)$ (the number of computational operations required for each temporal integration step increases as a cubic function of n). Since that ground-breaking work, a myriad of formulations and algorithms have been put forward by individuals whose interests lay in a wide variety of fields [1, 5–8, 12, 15, 20, 26, 28, 31]. Much of this has been done in an effort to develop more efficient, yet general simulation algorithms for multibody systems. In many of these situations, computational efficiency, which manifests itself in the form of computational speed, was of primary importance.

To this end, researchers have tried to improve the overall computational efficiency through vastly different approaches. Some researchers have pursued improved simulation speed through the use of different dynamic analysis methods for multibody systems, such as the methods based on Newton–Euler equations [4, 31], Lagrangian equations [15], Kane's equations [1, 7, 26], and variational methods [5, 6]. Others have pursued improved simulation speed through developing new more efficient underlying algorithms for the forward dynamics problem, such as $O(n^3)$, $O(n^2)$ [20, 31] and $O(n)$ algorithms [4, 5, 8, 12, 13]. While still other researchers have directed their attention to more efficient implementation of mathematical operations within existing formulations. Whichever avenue is pur-

sued, the dynamical equations of motions are most generally expressed in the form

$$\underline{u} = \underline{C}(\underline{q}, t) \cdot \dot{\underline{q}} + \underline{D}(\underline{q}), \tag{1a}$$

$$\underline{M}(\underline{q}, t; \underline{p}) \cdot \dot{\underline{u}} + \underline{\Phi}^{T}_{,\underline{q}}(\underline{q}, t; \underline{p}) \cdot \underline{\lambda} = \underline{K}(\underline{q}, \dot{\underline{q}}, t; \underline{p}), \tag{1b}$$

$$\underline{\Phi}(\underline{q}, t; \underline{p}) = \underline{0}. \tag{1c}$$

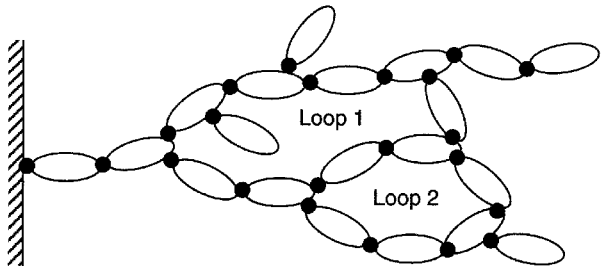
In each of these sets of equations $\underline{q}$ represents the set of n system generalized coordinates, with associated first time derivatives $\dot{\underline{q}}$. The matrix $\underline{C}$ appearing in Equation (1a) is a user specified invertible matrix which maps the set $\dot{\underline{q}}$ into the generalized speeds [18] $\underline{u}$, which are velocity level quasi-coordinates which can facilitate the characterization of the motion of the system. By comparison, matrix $\underline{D}$ in Equation (1a) is that portion of the definition of $\underline{u}$ which results from specified/prescribed motions which appear within the system. The matrix $\underline{M}$ is termed the system 'mass matrix' (though the system need not be purely mechanical in nature), while $\underline{K}$ is a column matrix containing contributions of all forcing terms, body loads, as well as centripetal and Coriolis acceleration inertia load contributions. The quantity $\underline{\Phi}_{,\underline{q}}$ is the constraint Jacobian associated with the partial derivative of the m independent algebraic constraint equations represented by (1c) with respect to $\underline{q}$.

In Equations (1) the state variables may be redundant, often representing position, orientation, temperature, pressure, voltage, etc. and their respective derivatives. If a redundancy exists, then generalized constraint 'forces' $\underline{\lambda}$ must be applied to the system equations of motion (1b) to enforce the algebraic constraint equations (1c).

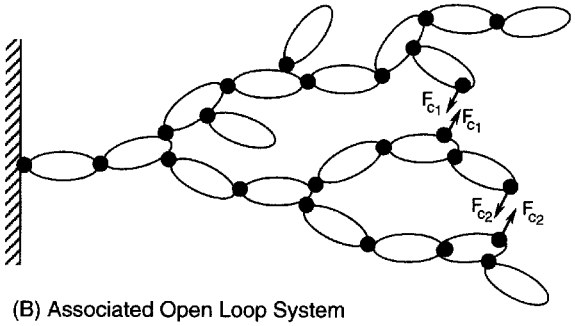
When traditional state-space dynamic analysis formulations are employed a computational cost of $O(n^3)$, is inevitable to produce $\underline{M}$ explicitly, if special methods are not used. If simple recursive relationships are observed, then the cost of generating $\underline{M}$ can be reduced to $O(n^2)$ [20, 25]. However, an additional $O(n^3)$ cost is incurred for the mass matrix decomposition required in solving Equation (1b) for the unknown state derivatives $\dot{\underline{u}}$ by direct methods. This expense may be acceptable for a system involving small to modest n, but an $O(n^3)$ increase in computational cost can become prohibitively expensive for even a modest increase of n. Fortunately, this prohibitive operational order can be greatly reduced through the intelligent use of efficient (often lower computational order) dynamic formulations.

2. Standard Recursive 'O(n)' Analysis for Constrained Systems

When analyzing closed loop systems (systems containing kinematic loops) with most state-space $O(n)$ formulations, the closed loop system is first converted to an open tree system. This is accomplished by cutting the closed loops at specified



(A) General Multibody System



(B) Associated Open Loop System

Figure 1. General closed loop system schematic and associated tree system.

joints, as indicated in Figure 1. At this time a unique path exits from each body to every other body in the system. To insure that the new tree system still behaves as the original closed loop system, explicit equal and opposite constraint forces $\underline{\mathbf{f}_c}$ are added to each side of the cut joints, and the equations of motions are augmented by the companion set of algebraic constraint equations (1c) which must now be satisfied.

2.1. MATHEMATICAL PRELIMINARIES

To aid in the subsequent development, consider the notation associated with the description of an arbitrary set of interconnected rigid bodies shown in Figure 2. For this system, proximal (parent) body Pr[k] is connected to its child body k through joint-k, via joint points k^- and k^+ which reside in bodies Pr[k] and k, respectively. Similarly, the *distal* (child) bodies of body k are given as members of the set of bodies Dist[k]. The position vector $\mathbf{s}^k$ locates joint-k relative to the mass center of body Pr[k], while the position vector $\mathbf{r}^k$ locates the mass center of body k with respect to the outboard end of this same joint. It will also prove convenient to

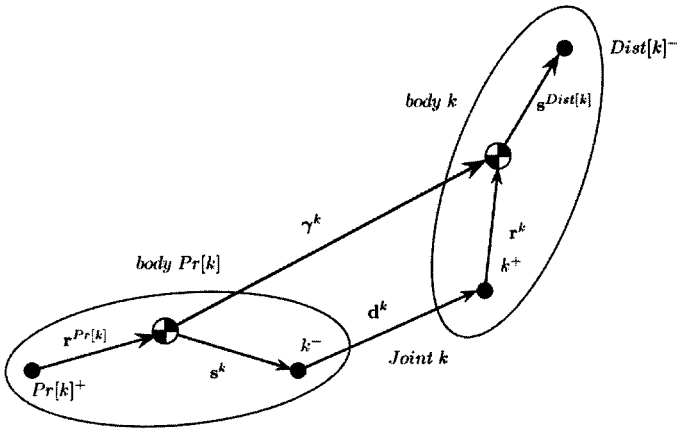


Figure 2. Notation associated with an arbitrary set of interconnected rigid bodies.

describe the position of child mass center k^* relative to proximal mass center $\mathrm{Pr}[k]^*$ by the vector $\boldsymbol{\gamma}^k$.

The angular velocity of any body k with respect to the Newtonian reference frame N, and velocity of its associated mass center k^* may always be written in terms of the generalized speeds as

$$\boldsymbol{\omega}^k = \sum_{r=1}^{n} \boldsymbol{\omega}_r^k u_r + \boldsymbol{\omega}_t^k \tag{2}$$

and

$$\mathbf{v}^{k^*} = \sum_{r=1}^{n} \mathbf{v}_r^{k^*} u_r + \mathbf{v}_t^{k^*}. \tag{3}$$

In these expressions $\boldsymbol{\omega}_r^k$ and $\mathbf{v}_r^{k^*}$ are termed the *r-th partial angular velocity of body k* and *r-th partial velocity of point* k^*, in N, respectively. These quantities may be thought of as basis vectors for the space of admissible system velocities and angular velocities, while the associated generalized speeds are the velocity space measure numbers. Additionally, the terms $\boldsymbol{\omega}_t^k$ and $\mathbf{v}_t^{k^*}$ appearing in Equations (2–3), are referred to as the *angular velocity remainder term* of body k and *velocity remainder term* of point k^*, in N, respectively. These quantities are most often associated with specified/prescribed motion, and thus are not associated with the time derivative of a system degree of freedom.

When deriving this method it is often convenient to express quantities in a scalar matrix, as opposed to a tensor (vector and dyadic) form. For this purpose an arbitrary vector ϑ^k will be represented in matrix form as $\underline{\vartheta^k}$, which is associated with the local dextral orthogonal unit vectors $\hat{k}_1$, $\hat{k}_2$, $\hat{k}_3$, fixed in body k. One may then define the *velocity*, *partial velocity*, and *velocity remainder term* matrices as

$$\underline{\mathcal{V}}^k = \begin{bmatrix} \underline{\omega}^k \\ \underline{v}^{k^*} \end{bmatrix}, \quad \underline{\mathcal{P}}_r^k = \begin{bmatrix} \underline{\omega}_r^k \\ \underline{v}_r^{k^*} \end{bmatrix}, \quad \text{and} \quad \underline{\mathcal{V}}_t^k = \begin{bmatrix} \underline{\omega}_t^k \\ \underline{v}_t^{k^*} \end{bmatrix}. \tag{4}$$

With these matrices so defined, Equations (2) and (3) may be expressed as

$$\underline{\mathcal{V}}^k = \underline{\bar{\mathcal{V}}}^k + \underline{\mathcal{V}}_t^k = \sum_{r=1}^{n} \underline{\mathcal{P}}_r^k u_r + \underline{\mathcal{V}}_t^k. \tag{5}$$

One can similarly represent the generalized acceleration matrix of an arbitrary body k as defined in previous works [1, 2], as

$$\underline{\mathcal{A}}^k = \begin{bmatrix} {}^N\underline{\alpha}^k \\ {}^N\underline{a}^{k^*} \end{bmatrix}, \tag{6}$$

where $\underline{\mathcal{A}}^k$ may also be divided into two portions. One is $\underline{\bar{\mathcal{A}}}^k$, which contains all terms which are explicit in the unknown state derivatives $\underline{\dot{u}}$ and the other is the *acceleration remainder term* $\underline{\mathcal{A}}_t^k$, which represents all of the other acceleration terms (and may be calculated directly from the system state), giving

$$\underline{\mathcal{A}}^k = \underline{\bar{\mathcal{A}}}^k + \underline{\mathcal{A}}_t^k. \tag{7}$$

2.2. RECURSIVE KINEMATIC RELATIONSHIPS

With the generalized velocity, generalized acceleration, and generalized acceleration remainder term matrices so represented, it is possible to compactly represent the recursive relationships necessary for determining all system kinematic quantities. As has been demonstrated in [1, 2] we have

$$\underline{\mathcal{V}}^k = \left[(\underline{\mathcal{S}}^k)^T \underline{\bar{\mathcal{V}}}^{\Pr[k]} + \underline{\mathcal{P}}_k^k \underline{u}_k \right] + \underline{\mathcal{V}}_t^k, \tag{8}$$

and

$$\underline{\mathcal{A}}^k = \left[(\underline{\mathcal{S}}^k)^T \underline{\bar{\mathcal{A}}}^{\Pr[k]} + \underline{\mathcal{P}}_k^k \underline{\dot{u}}_k \right] + \underline{\mathcal{A}}_t^k. \tag{9}$$

The quantity $\underline{\mathcal{S}}^k$ appearing in Equations (8–9) is the basis consistent linear transformation matrix

$$\underline{\mathcal{S}}^k = \begin{bmatrix} \underline{\mathcal{C}}^k & \underline{\mathcal{C}}^k \underline{\gamma}_\times^k \\ \underline{0} & \underline{\mathcal{C}}^k \end{bmatrix}_{6\times 6}. \tag{10}$$

Within this expression $\underline{\mathcal{C}}^k \equiv {}^{\Pr[k]}\underline{\mathcal{C}}^k$ is the direction cosine matrix which relates the body k basis vectors to those fixed in its parent body $\Pr[k]$; $\underline{0}$ is a 3×3 zero matrix; and $\underline{\gamma}_\times^k$ is the skew symmetric matrix equivalent to the vector cross product operation $\boldsymbol{\gamma}^k \times$. The *shift matrix* transformation $\underline{\mathcal{S}}^k$ converts a system of forces and moments acting through the center of mass of k, to an equivalent force system,

acting though a point of k which is instantaneously coincident with the center of mass of Pr[k].

At this time, it is also convenient to define the body k *generalized inertia* $\underline{\mathcal{I}}^k$ and the body k *generalized force* $\underline{\mathcal{F}}^k$ matrices

$$\underline{\mathcal{I}}^k = \begin{bmatrix} \underline{I}^{k/k^*} & \underline{0} \\ \underline{0} & \underline{M}^k \end{bmatrix}_{6\times 6}, \tag{11}$$

$$\underline{\mathcal{F}}^k = \begin{bmatrix} \underline{T}^k - (\underline{I}^{k/k^*}\,\underline{\alpha}_t^k + \underline{\omega}_\times^k\,\underline{I}^{k/k^*}\,\underline{\omega}^k) \\ \underline{R}^k - \underline{M}^k\,\underline{a}_t^k \end{bmatrix}_{6\times 1}. \tag{12}$$

Within these expressions, $\underline{I}^{k/k^*}$ is the 3×3 central inertia matrix of body k, and $\underline{M}^k$ is the diagonal translational mass matrix of this same body. By comparison $\underline{T}^k$ and $\underline{R}^k$ represent the resultant force system of all moments and forces, respectively, acting on body k through its center of mass k^*.

2.3. TRIANGULARIZATION OF EQUATIONS AND STANDARD 'O(N)' TREATMENT OF CONSTRAINTS

In most state space order-n methods the reduction in the computational order of the overall cost of determining values for the unknown state derivatives $\underline{\dot{u}}$ is largely accomplished by avoiding the formulation of the coupled set of equations of motion. The approach further reduces computational burden because it eliminates the need to subsequently decompose and solve these equations for their unknown state derivatives.

Avoiding the formulation of the explicit coupled system of equations is to a significant part accomplished through the segregation of accelerations into the *known* state dependent portion $\underline{\mathcal{A}}_t^k$ and that portion which is explicit (and linear) in the unknown state derivatives, $\underline{\bar{\mathcal{A}}}^k$. In this manner, unknown state derivatives $\underline{\dot{u}}$ can be isolated and manipulated (triagularized) as the equations are being formed. This allows the equations of motion to be produced in what is implicitly equivalent to a lower triangular form, where only the diagonal elements of the system triangularized mass matrix are explicitly calculated.

In a similar manner, the explicit constraint forces $\underline{F_c}$, associated with the cut joints which arise in the $O(n)$ treatment of closed loop systems, are recursively treated [1, 2, 6, 13, 17] and others. In these works, the closed loops are cut to produce an associated tree system. The explicit constraint forces, which manifest themselves in $\underline{\lambda}$ in Equation (1b), are then imposed to insure that the algebraic constraints (loop closure constraints) (1b), are satisfied.

Just as with the accelerations discussed above, the goal here is to segregate the applied forces $\underline{T}^k$ and $\underline{R}^k$, in (12), from the constraint forces $\underline{F_c}$. The applied forces $\underline{T}^k$ and $\underline{R}^k$, are directly determinable from the force element model and current system state values. By comparison, the constraint forces $\underline{F_c}$ are unknown, being linear in the unknown constraint force measure numbers $\underline{\lambda}$.

The segregation of constraint forces into dependent and independent portions can be accomplished by decomposing the unknown state derivatives $\underline{\dot{u}}$ as

$$\underline{\dot{u}} = \underline{\eta} + \underline{\zeta}\,\underline{\lambda}. \tag{13}$$

The quantity $\underline{\eta}$ appearing within (13), is that portion of $\underline{\dot{u}}$ which is independent of the constraint force measure numbers $\underline{\lambda}$, and is exactly the value obtained for $\underline{\dot{u}}$ if the constraints (1c) are not enforced. Thus, the quantity $\underline{\zeta}\,\underline{\lambda}$ may be thought of as a correction term to the unconstrained state derivatives.

The constraint forces have no influence on the kinematic calculations, so they only need be considered during and after the recursive triangularization portion of the $O(n)$ routines. The kinematic portion of the routine consists of the kinematic calculations working recursively outward from the system base body (system body without a parent) to the system terminal bodies (system bodies without children). Once the terminal bodies are reached, the topological direction with which the calculations proceed reverses, and the routine works recursively inward, from the terminal bodies to the base-body, forming generalized inertia forces, generalized applied forces, generalized constraint forces, and triangularizing the equations as they are formed.

Key quantities which aid in representation of the triangularization process are: the *articulated body inertia* and *articulated body force*, $\underline{\hat{\mathcal{I}}}^k$ and $\underline{\widehat{\mathcal{F}}}^k$, respectively [12]; the local triangularization matrix $\underline{\mathcal{T}}^k$; the triangularization matrix $\underline{\widehat{\mathcal{T}}}^k$; the triangularized mass matrix diagonal elements $\underline{\mathcal{M}}_k$; and the triangularized constraints associated with the i-th cut joint (closed loop) $\underline{\widehat{\mathcal{F}}}^k_{c_i}$.

Each of these quantities can be easily determined using the inward (from child to parent) recursive relationships

$$\underline{\hat{\mathcal{I}}}^k = \underline{\mathcal{I}}^k + \sum_{j \in \mathrm{Dist}[k]} \underline{\mathcal{T}}^j \underline{\hat{\mathcal{I}}}^j (\underline{\mathcal{S}}^j)^T, \tag{14}$$

$$\underline{\widehat{\mathcal{F}}}^k = \underline{\mathcal{F}}^k + \sum_{j \in \mathrm{Dist}[k]} \underline{\mathcal{T}}^j \underline{\widehat{\mathcal{F}}}^j, \tag{15}$$

$$\underline{\widehat{\mathcal{T}}}^{\mathrm{Pr}[k]} = \underline{\mathcal{T}}^k \underline{\widehat{\mathcal{T}}}^k, \tag{16}$$

$$\underline{\widehat{\mathcal{F}}}^k_{c_i} = \underline{\widehat{\mathcal{T}}}^k_i (\underline{E}_i \underline{\lambda}_i), \tag{17}$$

with

$$\underline{\mathcal{T}}^k = \underline{\mathcal{S}}^k [\underline{U} - \underline{\hat{\mathcal{I}}}^k \underline{\mathcal{P}}^k_k \underline{\mathcal{M}}^{-1}_k (\underline{\mathcal{P}}^k_k)^T], \tag{18}$$

and

$$\underline{\mathcal{M}}_k = (\underline{\mathcal{P}}^k_k)^T \underline{\hat{\mathcal{I}}}^k \underline{\mathcal{P}}^k_k, \tag{19}$$

where $\underline{U}$ appearing in (18) is a 6×6 identity matrix, and the quantity $\underline{E}_i$ appearing in (17) is a $6 \times m_i$ *expansion* matrix which converts the m_i unknown constraint load measure numbers in $\underline{\lambda}_i$, associated with the i-th cut joint ($i = 1, 2, \ldots, n_L$), to the associated constraint forces $\underline{f}_{c_i}$ and moments $\underline{t}_{c_i}$. Specifically, $\underline{E}_i$ provides the linear transformation such that

$$\underline{\mathcal{F}}_{c_i} = \begin{bmatrix} \underline{t}_{c_i} \\ \underline{f}_{c_i} \end{bmatrix} = \underline{E}_i \underline{\lambda}_i. \tag{20}$$

It can be shown [1, 2, 26] that with these *triangularized* quantities so defined, the equation of motion directly associated with the generalized speed u_k can be given as

$$(\underline{\mathcal{P}}_k^k)^T \left[\underline{\widehat{\mathcal{F}}}^k - \underline{\hat{\mathcal{I}}}^k \left(\underline{\bar{\eta}}^k + \sum_{i=1}^{n_L} \underline{\bar{\zeta}}_i^k \underline{\lambda}_i \right) + \sum_{i=1}^{n_L} \underline{\widehat{\mathcal{T}}}_i^k \underline{E}_i \underline{\lambda}_i \right] = 0, \tag{21}$$

where

$$\underline{\bar{\eta}}^k = (\underline{\mathcal{S}}^k)^T \underline{\bar{\eta}}^{\mathrm{Pr}[k]} + \underline{\mathcal{P}}_k^k \underline{\eta}^k, \tag{22}$$

and

$$\underline{\bar{\zeta}}_i^k = (\underline{\mathcal{S}}^k)^T \underline{\bar{\zeta}}_i^{\mathrm{Pr}[k]} + \underline{\mathcal{P}}_k^k \underline{\zeta}_i^k \quad (i = 1, \ldots, n_L). \tag{23}$$

2.4. FORWARD SUBSTITUTION AND STANDARD 'O(N)' TREATMENT OF CONSTRAINTS

The triangularization procedure works recursively inward from terminal bodies, towards the system base body (that body of the system which connects the system to the inertial reference frame) producing a system of equations which are implicitly equal to the triangularized equations. Once the system base body is reached, Equations (20–23) give that

$$\underline{\eta}^1 = \underline{\mathcal{M}}_1^{-1} (\underline{\mathcal{P}}_1^1)^T \underline{\widehat{\mathcal{F}}}^1, \tag{24}$$

and

$$\underline{\zeta}_i^1 = \underline{\mathcal{M}}_1^{-1} (\underline{\mathcal{P}}_1^1)^T \underline{\widehat{\mathcal{T}}}_i^1 \underline{E}_i. \tag{25}$$

All information associated with the right-hand sides of Equations (24) and (25) is known. Thus the process may now reverse its direction with respect to the system topology, and recursively perform a forward substitution for the determination of $\underline{\eta}^k$ and $\underline{\zeta}_i^k$. This forward substitution process is accomplished through the recursive use of

$$\underline{\bar{\eta}}^k = (\underline{\mathcal{S}}^k)^T \underline{\bar{\eta}}^{\mathrm{Pr}[k]} + \underline{\mathcal{P}}_k^k \underline{\eta}^k, \tag{26}$$

$$\underline{\eta}^k = \underline{\mathcal{M}}_k^{-1}(\underline{\mathcal{P}}_k^k)^T\left[\underline{\widehat{\mathcal{F}}}^k - \underline{\hat{\mathcal{I}}}^k(\underline{\mathcal{S}}^k)^T\underline{\bar{\eta}}^{\mathrm{Pr}[k]}\right], \tag{27}$$

and

$$\underline{\bar{\zeta}}_i^k = (\underline{\mathcal{S}}^k)^T\underline{\bar{\zeta}}_i^{\mathrm{Pr}[k]} + \underline{\mathcal{P}}_k^k\underline{\zeta}_i^k, \tag{28}$$

$$\underline{\zeta}_i^k = \underline{\mathcal{M}}_k^{-1}(\underline{\mathcal{P}}_k^k)^T\left[\underline{\widehat{\mathcal{T}}}_i^k\underline{E}_i - \underline{\hat{\mathcal{I}}}^k(\underline{\mathcal{S}}^k)^T\underline{\bar{\zeta}}_i^{\mathrm{Pr}[k]}\right]. \tag{29}$$

2.5. STANDARD 'O(N)' TREATMENT OF CONSTRAINTS

Unfortunately, determining $\underline{\eta}_{(n\times 1)}$ and $\underline{\zeta}_{(n\times m)}$ through use of Equations (24–29), Equation (13) still leaves the m unknowns $\underline{\lambda}_{(m\times 1)}$ to be determined. Thus one must make use of the m additional constraint equations

$$(\underline{\Phi}_{,\underline{q}})\underline{u} = \underline{\Psi} \tag{30}$$

which are obtained through the differentiation of Equation (1c) with respect to time. Differentiating this expression once more with respect to time yields

$$(\underline{\Phi}_{,\underline{q}})\dot{\underline{u}} = \underline{\psi}. \tag{31}$$

Substituting the expression for $\dot{\underline{u}}$, from Equation (13), into this expression produces

$$[\,\underline{\Phi}_{,\underline{q}}\,\underline{\zeta}\,]\underline{\lambda} = \{\underline{\psi} - \underline{\Phi}_{,\underline{q}}\,\underline{\eta}\}, \tag{32}$$

which one may solve to obtain the constraint load measure numbers $\underline{\lambda}$. These values for $\underline{\lambda}$ are in turn substituted back into expression (13), yielding the numerical values for $\dot{\underline{u}}$.

2.6. DIFFICULTIES WITH THE STANDARD 'O(N)' TREATMENT OF CONSTRAINTS

If one now looks more closely at the steps involved with the determination of the specific quantities $\underline{\eta}$, $\underline{\zeta}$, and $\underline{\lambda}$, which are used in determining the unknown state derivatives $\dot{\underline{u}}$ it can be seen that: The determination of $\underline{\eta}$ requires $O(n)$ operations; $\underline{\zeta}$ requires $O(nm)$ operations; and $\underline{\lambda}$ requires $O(nm + nm^2 + m^3)$ operations. As a result, the overall cost associated with the determination of $\dot{\underline{u}}$ with these *standard* so called '$O(n)$' approaches is actually $O(n + nm + nm^2 + m^3)$.

It should be noted that this cost does not consider the added computation, which may be necessary for some systems, to stabilize constraint violation error growth. Methods which may be considered in this regard are *Baumgarte's method* [9], *Coordinate Partitioning* [32, 33], or other forms of constraint error stabilization [1, 2, 22], each of which can be shown to contribute an additional $O(nm^2 + m^3)$ operations to the overall computational cost.

Thus it is apparent that these so-called '$O(n)$' approaches for closed loop systems may perform very well in situations where both $n \gg 1$ and $n \gg m$. However, if n and m are comparable in size (i.e. m is $O(n)$), then these '$O(n)$' state space methods actually yield $O(n^3)$ performance, and will often be out performed by other more traditional $O(n^3)$ methods.

Difficulties which arise in the application of the standard state space '$O(n)$' approach to moderately ($m \sim n$) constrained, or heavily ($m > n$) constrained systems have several sources. First, using this more standard $O(n)$ approach, the m_i unknown constraint load measure numbers $\underline{\lambda}$, manifesting themselves in $\mathcal{F}_{c_i}$ ($i = 1, \ldots, n_L$) are each individually passed (operated on) by each *ancestral body* (i.e. every body which is in the topological path between the application point of the constraint load and the system base body). This results in an $O(nm)$ expense for calculating these constraint force related terms within the equations of motion (13). It would be desirable to reduce, if not eliminate the need to operate on all constraint load unknowns by so many of the system bodies.

Second, the generation of the system level constraint Jacobian $\underline{\Phi}_{,\underline{q}}$, requires potentially $O(nm)$ operations. Worse still, the matrix manipulations, and decomposition and solve operations associated with the the determination of $\underline{\lambda}$, in Equations (31) and (32) requires $O(nm + nm^2 + m^3)$ operations. Again, it would be desirable to eliminate the explicit formulation of Equations (31) and (32), and subsequent solution for $\underline{\lambda}$.

Third, as indicated above, a method such as coordinate partitioning, for reducing constraint violation error is required with an additional computational burden of $O(nm + nm^2 + m^3)$. This cost can be significantly reduced by effectively applying the method in a more efficient manner, recursively employing it in the original generation of the constrained equations of motion.

3. Fully Recursive Coordinate Reduction for Closed Loop Systems

The primary computational efficiency gains associated with the new $O(n + m)$ method presented in this paper, relative to other $O(n)$ closed loop approaches [1–3, 6, 7, 13] are realized through the local and kinematic manner in which the loops are treated. To aid in this development, consider the generic closed loop i ($i = 1, \ldots, n_L$), shown in Figure 3. This loop development involves many symbols and expressions which are extremely similar in form and meaning to those which are associated with the global treatment (description) of the system. For this reason, all quantities which are specifically associated with the treatment of closed loops in this development will be denoted by a subscript i (e.g. k represents body-k of the system, while k_i represents the body-k of the close loop i under consideration).

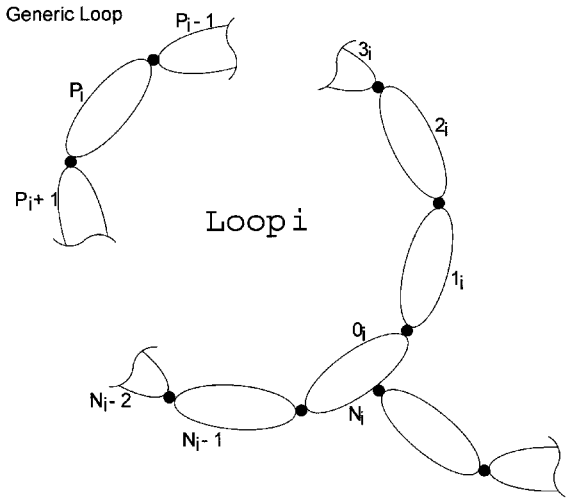


Figure 3. Local numbering scheme of generic loop.

3.1. LOCAL CLOSED LOOP KINEMATICS

The process begins by selecting the body which branches to form the i-th closed loop to act as the loop sub-system's primary reference frame. This body is locally referred to as the *loop base body* and is given the local body number 0_i. The bodies of the loop are then locally numbered consecutively, up through local body number n_i. Where n_i is the number of bodies which form the i-th closed loop and the last body n_i is a massless copy (or *phantom*) of the loop base body. Figure 4 illustrates the tree representation of the generic loop of Figure 3.

Within the closed loop the recursive velocity and acceleration relationships, (8) and (9) still apply. At the terminal end of the loop, the phantom body n_i must then have zero velocity and acceleration relative to the reference frame (body) 0_i. Thus, from Equation (8) we have

$$ {}^{0_i}\underline{\mathcal{V}}^{n_i} = (\underline{\mathcal{S}}^{n_i})^T \; {}^{0_i}\bar{\underline{\mathcal{V}}}^{n_i-1} + \underline{\mathcal{P}}^{n_i}_{n_i} u_{n_i} + {}^{0_i}\underline{\mathcal{V}}_t^{n_i} = \underline{0}. \tag{33} $$

Pre-multiplying (33) through by $(\mathcal{P}^{n_i}_{n_i})^T$ and then solving for the dependent generalized speeds $\underline{u}_{n_i}$ yields

$$ \underline{u}_{n_i} = -[(\underline{\mathcal{P}}^{n_i}_{n_i})^T \underline{\mathcal{P}}^{n_i}_{n_i}]^{-1}\big\{(\underline{\mathcal{P}}^{n_i}_{n_i})^T[(\underline{\mathcal{S}}^{n_i})^T \; {}^{0_i}\bar{\underline{\mathcal{V}}}^{n_i-1} + {}^{0_i}\underline{\mathcal{V}}_t^{n_i}]\big\}. \tag{34} $$

Similarly, from Equations (9) and (32), we have at the acceleration level

$$ \dot{\underline{u}}_{n_i} = -[(\underline{\mathcal{P}}^{n_i}_{n_i})^T \underline{\mathcal{P}}^{n_i}_{n_i}]^{-1}\big\{(\underline{\mathcal{P}}^{n_i}_{n_i})^T[(\underline{\mathcal{S}}^{n_i})^T \; {}^{0_i}\bar{\underline{\mathcal{A}}}^{n_i-1} + {}^{0_i}\underline{\mathcal{A}}_t^{n_i}]\big\}. \tag{35} $$

Appropriately substituting (34) into (33) produces

$$ \underline{\tau}^{n_i}[(\underline{\mathcal{S}}^{n_i})^T \; {}^{0_i}\bar{\underline{\mathcal{V}}}^{n_i-1} + {}^{0_i}\underline{\mathcal{V}}_t^{n_i}] = \underline{0}, \tag{36} $$

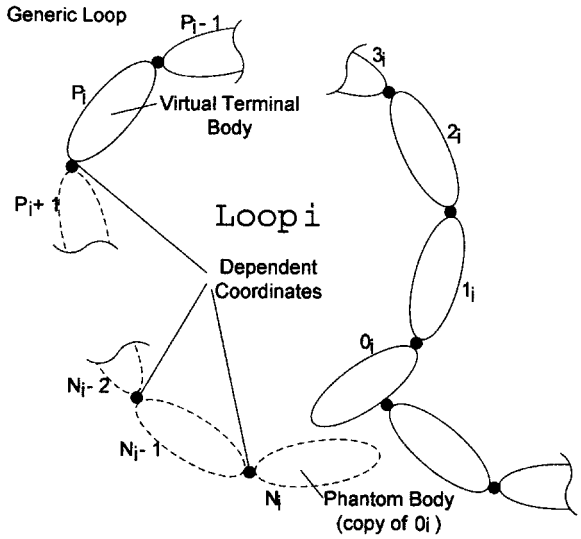


Figure 4. Tree representation of the generic loop (Figure 3) using a phantom body.

with

$$\underline{\Delta}^{n_i} = (\underline{\mathcal{P}}_{n_i}^{n_i})^T \underline{\mathcal{P}}_{n_i}^{n_i}, \tag{37}$$

and

$$\underline{\tau}^{n_i} = \underline{U} - \underline{\mathcal{P}}_{n_i}^{n_i} (\underline{\Delta}^{n_i})^{-1} (\underline{\mathcal{P}}_{n_i}^{n_i})^T. \tag{38}$$

Continuing the process, we have

$$\underline{\tau}^{n_i} \left[(\underline{\mathcal{J}}^{n_i})^T \left\{ (\underline{\mathcal{J}}^{n_i-1})^T \, {}^{0_i}\underline{\bar{\mathcal{V}}}^{n_i-2} + \underline{\mathcal{P}}_{n_i-1}^{n_i-1} \underline{u}_{n_i-1} \right\} + {}^{0_i}\underline{\mathcal{V}}_t^{n_i} \right] = \underline{0}. \tag{39}$$

Premultiplying this by $(\underline{\mathcal{P}}_{n_i-1}^{n_i-1})^T \underline{\mathcal{J}}^{n_i}$, then solving the resulting equation for $\underline{u}_{n_i-1}$ yields

$$u_{n_i-1} = -(\underline{\Delta}^{n_i-1})^{-1} (\underline{\mathcal{P}}_{n_i-1}^{n_i-1})^T \underline{\mathcal{J}}^{n_i} \underline{\tau}^{n_i} \left[(\underline{\mathcal{J}}^{n_i})^T (\underline{\mathcal{J}}^{n_i-1})^T \, {}^{0_i}\underline{\bar{\mathcal{V}}}^{n_i-2} + {}^{0_i}\underline{\mathcal{V}}_t^{n_i} \right], \tag{40}$$

where

$$\underline{\Delta}^{n_i-1} = (\underline{\mathcal{P}}_{n_i-1}^{n_i-1})^T \underline{\mathcal{J}}^{n_i} \underline{\tau}^{n_i} (\underline{\mathcal{J}}^{n_i})^T \underline{\mathcal{P}}_{n_i-1}^{n_i-1}. \tag{41}$$

Substituting (40) back into (39) yields

$$\left\{ \underline{\tau}^{n_i} - \underline{\tau}^{n_i} (\underline{\mathcal{J}}^{n_i})^T \underline{\mathcal{P}}_{n_i-1}^{n_i-1} (\underline{\Delta}^{n_i-1})^{-1} (\underline{\mathcal{P}}_{n_i-1}^{n_i-1})^T \underline{\mathcal{J}}^{n_i} \underline{\tau}^{n_i} \right\}$$
$$\times \left[(\underline{\mathcal{J}}^{n_i})^T (\underline{\mathcal{J}}^{n_i-1})^T \, {}^{0_i}\underline{\bar{\mathcal{V}}}^{n_i-2} + {}^{0_i}\underline{\mathcal{V}}_t^{n_i} \right] = \underline{0}. \tag{42}$$

Defining the following recursive relationships for intermediate quantities

$${}^{n_i}\underline{\mathcal{J}}^{n_i} = \underline{U} \quad \Rightarrow \quad {}^{k-1}\underline{\mathcal{J}}^{n_i} = {}^{k-1}\underline{\mathcal{J}}^{k} \, {}^{k}\underline{\mathcal{J}}^{n_i} = \underline{\mathcal{J}}^{k} \, {}^{k}\underline{\mathcal{J}}^{n_i}, \tag{43}$$

$$\underline{\tau}^{n_i+1} = \underline{U}$$

$$\Rightarrow \quad \underline{\tau}^{k} = \left[\underline{\tau}^{k+1} - \underline{\tau}^{k+1}({}^{k}\underline{\mathcal{J}}^{n_i})^T \underline{\mathcal{P}}^{k}_{k}(\underline{\Delta}^{k})^{-1}(\underline{\mathcal{P}}^{k}_{k})^T {}^{k}\underline{\mathcal{J}}^{n_i}\underline{\tau}^{k+1}\right], \tag{44}$$

$$\underline{\Delta}^{k} = (\underline{\mathcal{P}}^{k}_{k})^T {}^{k}\underline{\mathcal{J}}^{n_i}\underline{\tau}^{k+1}({}^{k}\underline{\mathcal{J}}^{n_i})^T \underline{\mathcal{P}}^{k}_{k}, \tag{45}$$

we write the recursive relationships

$${}^{0_i}\underline{\bar{\mathcal{V}}}^{k} = \underline{\delta}^{k}\ {}^{0_i}\underline{\bar{\mathcal{V}}}^{k-1} + \underline{\chi}^{k}\ {}^{0_i}\underline{\mathcal{V}}_{t}^{n_i}, \tag{46}$$

and

$$\underline{u}_k = -(\underline{\Delta}^{k})^{-1}(\underline{\mathcal{P}}^{k}_{k})^T\ {}^{k}\underline{\mathcal{J}}^{n_i}\underline{\tau}^{k+1}\left[({}^{k-1}\underline{\mathcal{J}}^{n_i})^T\ {}^{0_i}\underline{\bar{\mathcal{V}}}^{k-1} + {}^{0_i}\underline{\mathcal{V}}_{t}^{n_i}\right], \tag{47}$$

with

$$\underline{\chi}^{k} = -\underline{\mathcal{P}}^{k}_{k}(\underline{\Delta}^{k})^{-1}(\underline{\mathcal{P}}^{k}_{k})^T\ {}^{k}\underline{\mathcal{J}}^{n_i}\underline{\tau}^{k+1} \tag{48}$$

and

$$\underline{\delta}^{k} = (\underline{\mathcal{J}}^{k})^T + \underline{\chi}^{k}\ ({}^{k-1}\underline{\mathcal{J}}^{n_i})^T. \tag{49}$$

The process may be continued, allowing all terms associated with the dependent degrees of freedom to be expressed in terms of the independent generalized speeds $\underline{u}_{1_i}$–$\underline{u}_{p_i}$ and their derivatives, namely

$${}^{0_i}\underline{\bar{\mathcal{V}}}^{k} = \underline{\Gamma}^{k}\ {}^{0_i}\underline{\bar{\mathcal{V}}}^{p_i} + \Xi^{k}\ {}^{0_i}\underline{\mathcal{V}}_{t}^{n_i}, \tag{50}$$

and

$$\begin{aligned}\underline{u}_k \ &= \ -(\underline{\Delta}^{k})^{-1}(\underline{\mathcal{P}}^{k}_{k})^T\ {}^{k}\underline{\mathcal{J}}^{n_i}\underline{\tau}^{k+1} \\ &\times\ \left[\{({}^{k-1}\underline{\mathcal{J}}^{n_i})^T\ \underline{\Gamma}^{k}\}{}^{0_i}\underline{\bar{\mathcal{V}}}^{p_i} + \{({}^{k-1}\underline{\mathcal{J}}^{n_i})^T\ \underline{\Xi}^{k} + \underline{U}\}{}^{0_i}\underline{\mathcal{V}}_{t}^{n_i}\right],\end{aligned} \tag{51}$$

where

$$\underline{\Gamma}^{p_i} = \underline{U}, \quad \text{and} \quad \underline{\Gamma}^{k+1} = \underline{\delta}^{k+1}\underline{\Gamma}^{k}, \tag{52}$$

and

$$\underline{\Xi}^{p_i} = \underline{0}, \quad \text{and} \quad \underline{\Xi}^{k+1} = \underline{\delta}^{k+1}\underline{\Xi}^{k} + \underline{\chi}^{k+1}, \tag{53}$$

Similarly, at the acceleration level we have

$${}^{0_i}\underline{\bar{\mathcal{A}}}^{k} = \underline{\Gamma}^{k}\ {}^{0_i}\underline{\bar{\mathcal{A}}}^{p_i} + \underline{\Xi}^{k}\ {}^{0_i}\underline{\mathcal{A}}_{t}^{n_i}, \tag{54}$$

and

$$\begin{aligned}\underline{\dot{u}}_k \ &= \ -(\underline{\Delta}^{k})^{-1}(\underline{\mathcal{P}}^{k}_{k})^T\ {}^{k}\underline{\mathcal{J}}^{n_i}\underline{\tau}^{k+1} \\ &\times\ \left[\{({}^{k-1}\underline{\mathcal{J}}^{n_i})^T\ \underline{\Gamma}^{k}\}{}^{0_i}\underline{\bar{\mathcal{A}}}^{p_i} + \{({}^{k-1}\underline{\mathcal{J}}^{n_i})^T\ \underline{\Xi}^{k} + \underline{U}\}{}^{0_i}\underline{\mathcal{A}}_{t}^{n_i}\right].\end{aligned} \tag{55}$$

3.2. LOCAL CLOSED LOOP KINETICS

At this point sufficient kinematic quantities are available that one may start the process of assembling the equations of motion associated with the constrained degrees of freedom within this loop.

It can be shown [1, 2, 26] that due to the use of relative coordinates, the equation of motion associated with degree of freedom p, which immediately precedes the last m bodies of a tree system may be written as

$$\sum_{k=p_i}^{p_i+m_i} (\underline{\mathcal{P}}^k_{p_i})^T (\underline{\mathcal{I}}^k \underline{\bar{\mathcal{A}}}^k - \underline{\mathcal{F}}^k) = \underline{0}. \tag{56}$$

In situations involving nonholonomic constraints, this equation becomes

$$\sum_{k=p_i}^{p_i+m_i} (\underline{\widetilde{\mathcal{P}}}^k_{p_i})^T (\underline{\mathcal{I}}^k \underline{\bar{\mathcal{A}}}^k - \underline{\mathcal{F}}^k) = \underline{0}, \tag{57}$$

where $\widetilde{\mathcal{P}}^k_{p_i}$ is the *nonholonomic spatial partial velocity* of body k associated with independent generalized speed p_i. These nonholomonic partial velocities are those which result once the nonholonomic constraints have been used to eliminate the dependent generalized speeds and their derivatives from all expressions. Making use of Equation (5) and the definitions for $\underline{\mathcal{P}}^k_{p_i}$ (4), ${}^{0_i}\underline{\mathcal{S}}^k$ (43), and ${}^{0_i}\underline{\bar{\mathcal{V}}}^k$ (50) these nonholonomic spatial partial velocities are given by

$$\begin{aligned}
\underline{\widetilde{\mathcal{P}}}^k_{p_i} &= \frac{\partial \underline{\bar{\mathcal{V}}}^k}{\partial u_{p_i}} \\
&= \frac{\partial}{\partial u_{p_i}} \left[({}^{0_i}\underline{\mathcal{S}}^k)^T \underline{\bar{\mathcal{V}}}^{0_i} + {}^{0_i}\underline{\bar{\mathcal{V}}}^k \right] \\
&= \frac{\partial}{\partial u_{p_i}} \left[({}^{0_i}\underline{\mathcal{S}}^k)^T \underline{\bar{\mathcal{V}}}^{0_i} + (\underline{\Gamma}^k \, {}^{0_i}\underline{\bar{\mathcal{V}}}^{p_i} + \Xi^k \, {}^{0_i}\underline{\mathcal{V}}^{n_i}) \right] \\
&= \underline{\Gamma}^k \, \underline{\mathcal{P}}^{p_i}_{p_i}.
\end{aligned} \tag{58}$$

Also, it is helpful to note that the accelerations $\underline{\bar{\mathcal{A}}}^k$ appearing in Equation (57) may be rewritten in terms of $\underline{\bar{\mathcal{A}}}^{0_i}$ and ${}^{0_i}\underline{\bar{\mathcal{A}}}^k$. Specifically, for the situation where $0_i \leq k < p_i$

$$\underline{\bar{\mathcal{A}}}^k = ({}^{0_i}\underline{\mathcal{S}}^k)^T \underline{\bar{\mathcal{A}}}^{0_i} + {}^{0_i}\underline{\bar{\mathcal{A}}}^k. \tag{59}$$

Substituting relations (54) into (59), with this result in turn substituted into equation (57) along with (58) yields

$$\sum_{k=p_i}^{p_i+m_i} (\underline{\widetilde{\mathcal{P}}}^{p_i}_{p_i})^T (\underline{\mathcal{I}}^k \underline{\bar{\mathcal{A}}}^k - \underline{\mathcal{F}}^k) = \underline{0}$$

$$\Rightarrow \quad (\underline{\mathscr{P}}_{p_i}^{p_i})^T \sum_{k=p_i}^{p_i+m_i} (\underline{\Gamma}^k)^T \left(\underline{\mathscr{I}}^k \underline{\bar{\mathscr{A}}}^k - \underline{\mathscr{F}}^k\right) = \underline{0}$$

$$\Rightarrow \quad (\underline{\mathscr{P}}_{p_i}^{p_i})^T \sum_{k=p_i}^{p_i+m_i} (\underline{\Gamma}^k)^T \left\{\underline{\mathscr{I}}^k \left[({}^{0_i}\underline{\mathscr{S}}^k)^T \underline{\bar{\mathscr{A}}}^{0_i} + {}^{0_i}\underline{\bar{\mathscr{A}}}^k\right] - \underline{\mathscr{F}}^k\right\} = \underline{0}$$

$$\Rightarrow \quad (\underline{\mathscr{P}}_{p_i}^{p_i})^T \sum_{k=p_i}^{p_i+m_i} (\underline{\Gamma}^k)^T \left\{\underline{\mathscr{I}}^k \left[({}^{0_i}\underline{\mathscr{S}}^k)^T \underline{\bar{\mathscr{A}}}^{0_i} + (\underline{\Gamma}^k \; {}^{0_i}\underline{\bar{\mathscr{A}}}^{p_i} + \Xi^k \; {}^{0_i}\underline{\mathscr{A}}_t^{n_i})\right] - \underline{\mathscr{F}}^k\right\} = \underline{0}$$

$$\Rightarrow \quad (\underline{\mathscr{P}}_{p_i}^{p_i})^T \left\{ \left[\sum_{k=p_i}^{p_i+m_i} (\underline{\Gamma}^k)^T \underline{\mathscr{I}}^k ({}^{0_i}\underline{\mathscr{S}}^k)^T\right] \underline{\bar{\mathscr{A}}}^{0_i} + \left[\sum_{k=p}^{p+m_i} (\underline{\Gamma}^k)^T \underline{\mathscr{I}}^k \underline{\Gamma}^k\right] {}^{0_i}\underline{\bar{\mathscr{A}}}^{p_i} \right.$$
$$\left. + \left[\sum_{k=p_i}^{p_i+m_i} (\underline{\Gamma}^k)^T \underline{\mathscr{I}}^k \underline{\Xi}^k\right] {}^{0_i}\underline{\mathscr{A}}_t^{n_i} - \left[\sum_{k=p}^{p+m_i} (\underline{\Gamma}^k)^T \underline{\mathscr{F}}^k\right] \right\} = \underline{0}$$

$$\Rightarrow \quad (\underline{\mathscr{P}}_{p_i}^{p_i})^T \left[\underline{\hat{\mathscr{I}}}^{p_i;0_i} \; \underline{\bar{\mathscr{A}}}^{0_i} + {}^{0_i}\underline{\hat{\mathscr{I}}}^{p_i} \; {}^{0_i}\underline{\bar{\mathscr{A}}}^{p_i} - {}^{0_i}\underline{\hat{\mathscr{F}}}^{p_i}\right] = \underline{0}. \tag{60}$$

What is left is effectively a new *virtual* terminal body p_i, as indicated in Figure 4, which is represented by multiple inertia properties $\underline{\hat{\mathscr{I}}}^{p_i;0_i}$ and ${}^{0_i}\underline{\hat{\mathscr{I}}}^{p_i}$, as well as applied forces ${}^{0_i}\underline{\hat{\mathscr{F}}}^{p_i}$. Inspection of Equations (43–49), (52–54), and (57–60) indicate that the production of these quantities requires only $O(m_i)$ additional operations beyond those required for the recursive determination of kinematic quantities of the associated unconstrained system. This is due to the fact that the operations indicated by Equations (43–60), are recursive in nature and as a rule consist of a fixed number of multiplication and addition operations per body. Additionally, these virtual bodies have embedded within them all the affects of the m_i constraints associated with closed loop i ($i = 1, 2, \ldots, n_L$) at both the velocity and acceleration levels. Most importantly, these virtual bodies, and all of their associated properties may be treated effectively as unconstrained bodies in the general unconstrained $O(n)$ state-space formulation.

Indeed, the terms $\underline{\hat{\mathscr{I}}}^{p_i;0_i}$, ${}^{0_i}\underline{\hat{\mathscr{I}}}^{p_i}$, and ${}^{0_i}\underline{\hat{\mathscr{F}}}^{p_i}$ may be recursively triagularized in much the same manner as the terms in the standard state-space $O(n)$ algorithms for unconstrained systems. Specifically, for $0_i < k \leq p_i$

$$(\underline{\mathscr{P}}_k^k)^T \left[\underline{\hat{\mathscr{I}}}^{k;0_i} \; \underline{\bar{\mathscr{A}}}^{0_i} + {}^{0_i}\underline{\hat{\mathscr{I}}}^k \; {}^{0_i}\underline{\bar{\mathscr{A}}}^k - {}^{0_i}\underline{\hat{\mathscr{F}}}^{k_i}\right] = \underline{0}, \tag{61}$$

with the recursive relationships

$$\underline{\hat{\mathscr{I}}}^{k-1;0_i} = \underline{\mathscr{I}}^{k-1}({}^{0_i}\underline{\mathscr{S}}^{k-1})^T + \underline{\mathscr{T}}^k \; \underline{\hat{\mathscr{I}}}^{k;0_i}, \tag{62}$$

$${}^{0_i}\underline{\hat{\mathscr{I}}}^{k-1} = \underline{\mathscr{I}}^{k-1} + \underline{\mathscr{T}}^k \; {}^{0_i}\underline{\hat{\mathscr{I}}}^k \; (\underline{\mathscr{S}}^k)^T, \tag{63}$$

$$\underline{{}^{0_i}\hat{\mathcal{F}}}^{k-1} = \mathcal{F}^{k-1} + \underline{\mathcal{T}}^k \; \underline{{}^{0_i}\hat{\mathcal{F}}}^k, \tag{64}$$

where $\underline{\mathcal{T}}^k$ is our triangularization operation matrix

$$\underline{\mathcal{T}}^k = \underline{\mathcal{S}}^k \left[\underline{U} - \underline{{}^{0_i}\hat{\mathcal{I}}}^k \, \underline{\mathcal{P}}^k_k (\underline{\mathcal{M}}_k)^{-1} (\underline{\mathcal{P}}^k_k)^T \right] \tag{65}$$

and

$$\underline{\mathcal{M}}_k = (\underline{\mathcal{P}}^k_k)^T \; \underline{{}^{0_i}\hat{\mathcal{I}}}^k \; \underline{\mathcal{P}}^k_k. \tag{66}$$

The process then continues recursively until body 0_i is reached, at which $\underline{{}^{0_i}\bar{\mathcal{A}}}^{0_i} = \underline{0}$ and the associated inertia term is dropped. At this point $\underline{\hat{\mathcal{I}}}^{k;0_i}$ and $\underline{{}^{0_i}\hat{\mathcal{F}}}^k$ become synonymous with $\underline{\hat{\mathcal{I}}}^k$ and $\underline{\hat{\mathcal{F}}}^k$, respectively, for $k = 0_i$. The procedure then continues as an unconstrained system, making use of Equations (14), (15), (19), (24), (26), and (27).

Once the base body is reached, the algorithm again works recursively outward, this time performing the recursive forward substitution. For the situation where the body k being considered lies inboard of body 0_i, the recursive relations are effectively those given by Equations (26) and (27)

$$\dot{\underline{u}}_k = \underline{\mathcal{M}}_k^{-1} (\underline{\mathcal{P}}^k_k)^T \left[\underline{{}^{0_i}\hat{\mathcal{F}}}^k - \underline{\hat{\mathcal{I}}}^k (\underline{\mathcal{S}}^k)^T \underline{\bar{\mathcal{A}}}^{\Pr[k]} \right], \tag{67}$$

and

$$\underline{\bar{\mathcal{A}}}^k = (\underline{\mathcal{S}}^k)^T \underline{\bar{\mathcal{A}}}^{\Pr[k]} + \underline{\mathcal{P}}^k_k \dot{\underline{u}}_k, \tag{68}$$

where

$$\underline{\bar{\mathcal{A}}}^0 = \underline{0}. \tag{69}$$

For the situations where the body k lies between bodies 0_i and p_i within a closed loop i (i.e. $0_i < k \leq p_i$), these forward substitution relations are modified to

$$\dot{\underline{u}}_k = \underline{\mathcal{M}}_k^{-1} (\underline{\mathcal{P}}^k_k)^T \left(\underline{{}^{0_i}\hat{\mathcal{F}}}^k - [\underline{\hat{\mathcal{I}}}^{k;0_i} \, \underline{\bar{\mathcal{A}}}^{0_i} + \underline{{}^{0_i}\hat{\mathcal{I}}}^k (\underline{\mathcal{S}}^k)^T \; \underline{{}^{0_i}\bar{\mathcal{A}}}^{\Pr[k]}] \right), \tag{70}$$

and

$$\underline{{}^{0_i}\bar{\mathcal{A}}}^k = (\underline{\mathcal{S}}^k)^T \underline{{}^{0_i}\bar{\mathcal{A}}}^{\Pr[k]} + \underline{\mathcal{P}}^k_k \dot{\underline{u}}_k. \tag{71}$$

The resulting Recursive Coordinate Reduction (RCR) procedure is fundamentally the same as the two kinematic formulations appearing in Stajskal and Valášek [29]. However it has been formulated independently within the frame work of Kane's equations and Anderson's $O(n)$ algorithm. And, as will be demonstrated, it is more general.

4. Results

In this section, the equations of Section 3 which describe a method for fully recursive coordinate reduction (RCR) is demonstrated with a simple planar four-bar

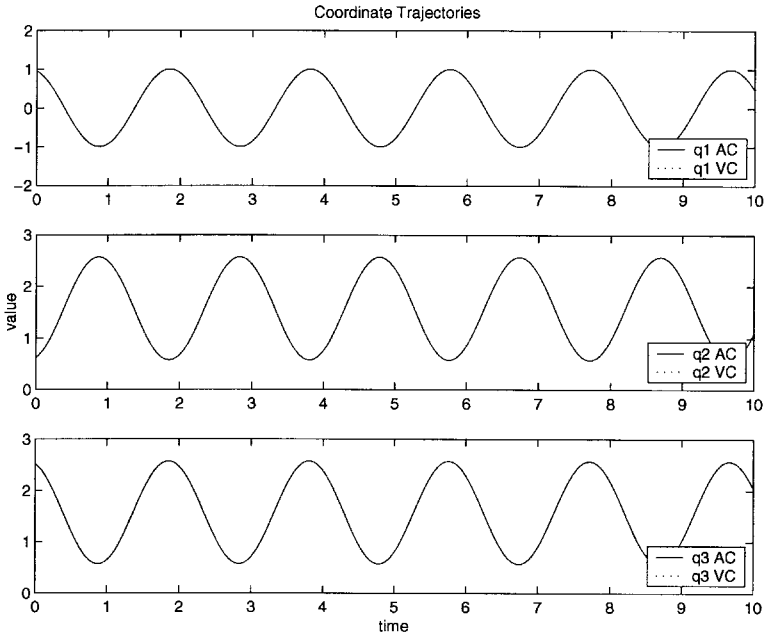


Figure 5. Four-bar linkage overlapping trajectories.

mechanism, a planar multi-loop redundant parallel five-bar mechanism, as well as three variations of a heavily constrained spatial ladder mechanism. The RCR is shown to exhibit both the expected solution for such mechanisms as well as superior constraint stability relative to the 'Traditional $O(n)$' constraint technique.

4.1. CONSTRAINT STABILITY COMPARISON

Both the traditional $O(n)$ constraint formulation and the new method of recursive coordinate reduction have been used to simulate a four-bar linkage. The result is the expected overlapping coordinate trajectories of Figure 5, which were obtained using a fourth order fixed step (0.01 seconds, no error check) Runge–Kutta integrator. However the displacement error associated with the acceleration level constraint (AC) enforcement of the more traditional $O(n)$ approach (if no additional form of constraint stabilization is used), is very different than that of the velocity level constraint (VC) enforcement which is inherent in the presented RCR formulation, as shown in Figure 6. One notes that both systems are started with identical initial conditions (displacement and velocity) which are accurate to nine decimal places, and the RCR method, which enforces the constraints to machine accuracy at the velocity level, demonstrates the expected result of a constant rate of drift in displacement at the ninth decimal place. By comparison, the traditional approach, which enforces the constraints only at the acceleration level, drifts with an expected

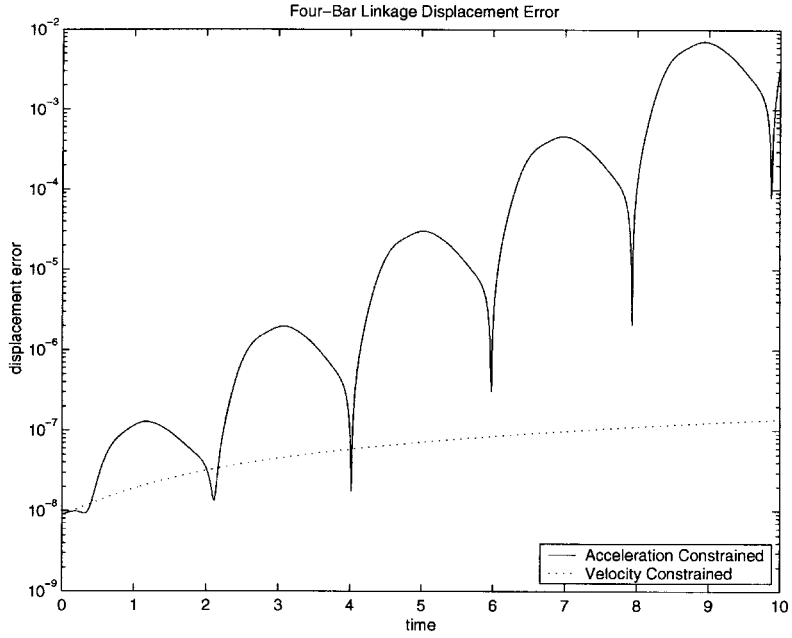


Figure 6. Four-bar linkage accuracy comparison.

logarithmic trend ($\sim$ quadratic in time), as was previously demonstrated by Park and Chiou [22].

The inclusion of superior constraint stability at no additional cost to the method is a significant improvement in itself.

4.2. PARALLEL MANIPULATORS

Although formulated for single independent (uncoupled) loops, the equations of Section 3 can be applied exactly as written to all forms of parallel manipulators. Parallel manipulators are structures which may be decomposed into several linkages (legs) extending from a common base and terminating at a common end-effector.

Examples of the resulting topology of arbitrary parallel manipulators are shown in Figure 7. These examples illustrate that the independent coordinates can all be assigned consecutively because all legs meet at the single end-effector which is initially located through the traversal of a series of independent coordinates. Any remaining independent degrees of freedom are then selected as the coordinates adjacent to the end effector, and the effects of the dependent coordinates are placed on the loops' local virtual terminal bodies p_i.

The planar redundant parallel five-bar manipulator of Figure 7 has been implemented with the Recursive Coordinate Reduction method. Figure 8 shows the

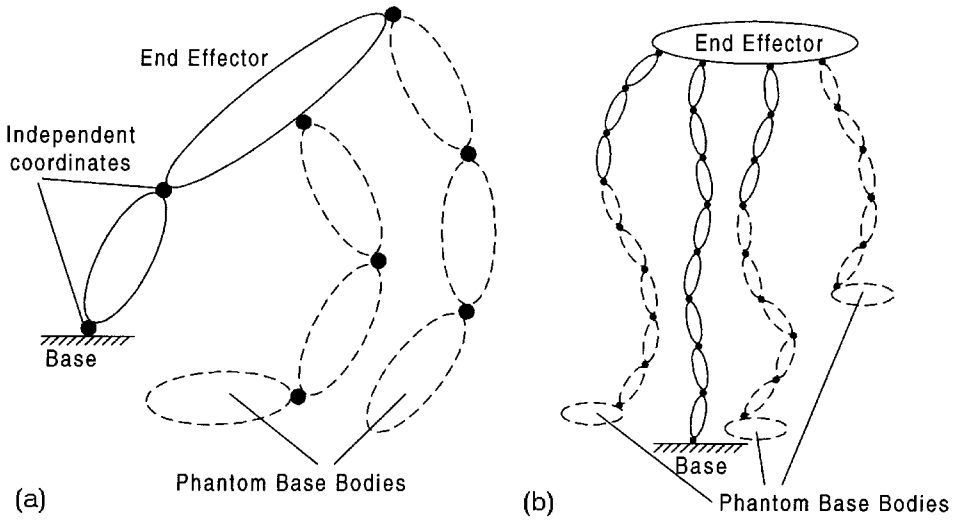


Figure 7. Examples of parallel manipulators. (a) A redundant parallel five-bar mechanism. (b) An under-actuated parallel mechanism.

trajectories of the independent coordinates compared with the results of an independent multibody package (AUTOLEV).

At this time it is important to separate the RCR method from the $O(n)$ formulation of Saha and Schiehlen [27] which applies strictly to nonredundant parallel (*legged*) manipulators. Saha and Schiehlen's formulation requires that a single independent coordinate be placed in each leg, thus the broader class of redundant and under-actuated parallel manipulators mentioned here are outside of the scope of the work. However, the extension to such systems *may* be possible, but the resulting formulation would be as fundamentally different in structure as the selection of independent coordinates.

4.3. OTHER HEAVILY CONSTRAINED SYSTEMS

Application of recursive coordinate reduction to heavily constrained systems is demonstrated through simulation of the *spatial ladder* systems shown in Figure 9. In these systems, the bodies are connected to ground by two degree of freedom Hooke's joints and interconnected in a ladder type formation by single degree of freedom revolute joints. An arbitrary example of such a system consisting of L coupled closed loops will involve; $N = 2L + 1$ bodies; L phantom bodies; $n = 2N = 4L + 2$ generalized coordinates; $m = 4L$ algebraic constraints; yet only have 2 degrees of freedom. This system represents a situation where previously developed dynamic simulation and analysis algorithms are likely to pay a high computational price both for the number of generalized coordinates n and the number of algebraic constraints m.

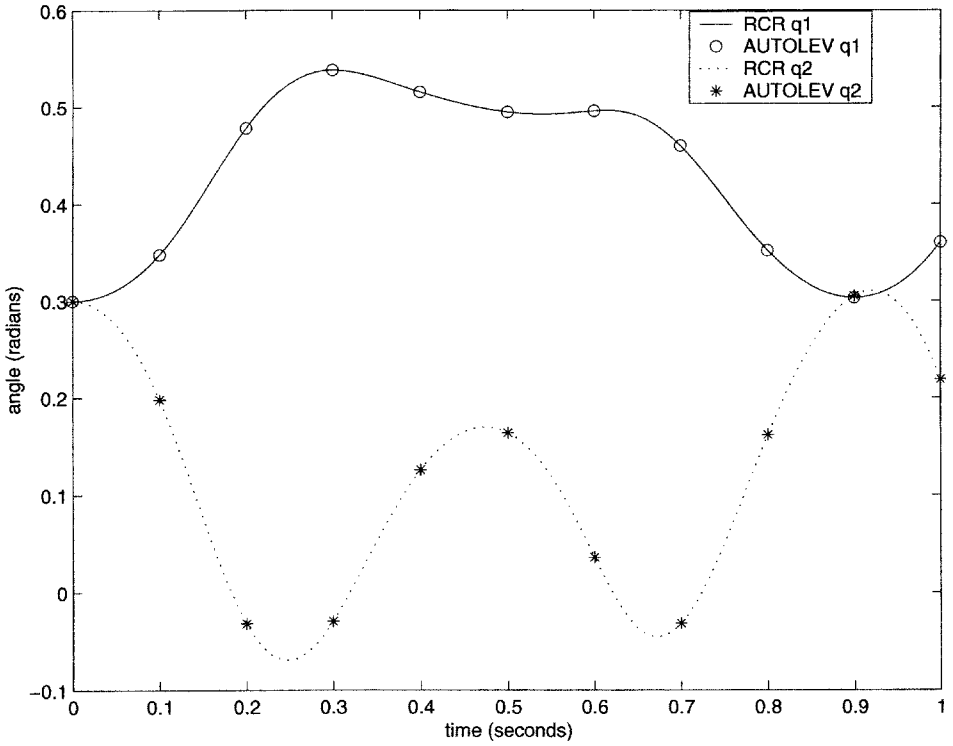


Figure 8. Redundant five-bar manipulator independent coordinate trajectories.

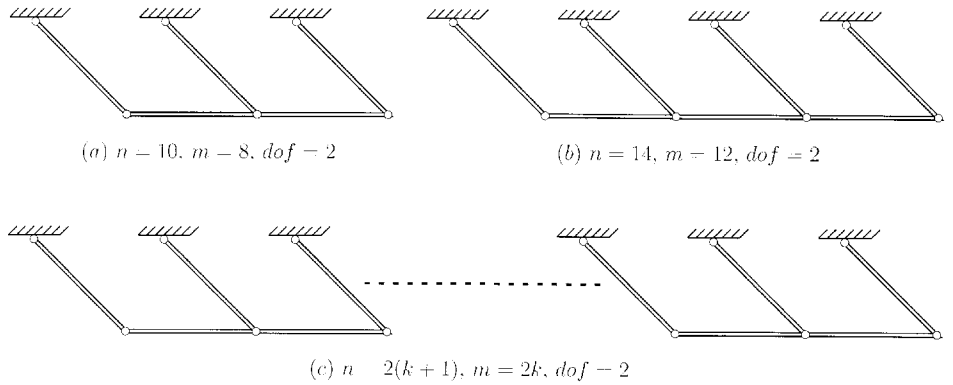


Figure 9. Schematic of a heavily constrained system.

Direct application of the previous equations to the ladder systems will actually result in a quadratic growth in the number of computations. This is due to the fact that each phantom body's (n_i's) kinematic loop quantities must be related and computed back to body p_i, which requires a linearly increasing effort for each additional cell (or ladder rung).

To avoid quadratic growth and maintain a linear formulation we note that for loops sharing a common base body (e.g. $0_i = 0_j$), the following relationships hold at the connection to the first body in a new cell:

$$
\begin{aligned}
{}^{0_j}\underline{\bar{\mathcal{A}}}^{k_j} &= \underline{\delta}^{k_j}\ {}^{0_j}\underline{\bar{\mathcal{A}}}^{\Pr[k_j]} + \underline{\chi}^{k_j}\ {}^{0_j}\underline{\mathcal{A}}_t^{n_j} && (72)\\
&= \underline{\delta}^{k_j}\ {}^{0_i}\underline{\bar{\mathcal{A}}}^{k_i} + \underline{\chi}^{k_j}\ {}^{0_j}\underline{\mathcal{A}}_t^{n_j} && (73)
\end{aligned}
$$

and

$$
{}^{0_i}\underline{\bar{\mathcal{A}}}^{k_i} = \underline{\Gamma}^{k_i}\ {}^{0_i}\underline{\bar{\mathcal{A}}}^{p_i} + \underline{\Xi}^{k_i}\ {}^{0_i}\underline{\mathcal{A}}_t^{n_i}. \tag{74}
$$

So we may write

$$
{}^{0_j}\underline{\bar{\mathcal{A}}}^{k_j} = \underline{\delta}^{k_j}\ (\underline{\Gamma}^{k_i}\ {}^{0_i}\underline{\bar{\mathcal{A}}}^{p_i} + \underline{\Xi}^{k_i}\ {}^{0_i}\underline{\mathcal{A}}_t^{n_i}) + \underline{\chi}^{k_j}\ {}^{0_j}\underline{\mathcal{A}}_t^{n_j}, \tag{75}
$$

or

$$
\begin{aligned}
{}^{0_j}\underline{\bar{\mathcal{A}}}^{k_j} &= \underline{\Gamma}^{k_j}\ {}^{0_i}\underline{\bar{\mathcal{A}}}^{p_i} + \underline{\Theta}^{k_j} + \underline{\chi}^{k_j}\ {}^{0_j}\underline{\mathcal{A}}_t^{n_j} && (76)\\
&= \underline{\Gamma}^{k_j}\ {}^{0_i}\underline{\bar{\mathcal{A}}}^{p_j} + \underline{\Theta}^{k_j} + \underline{\chi}^{k_j}\ {}^{0_j}\underline{\mathcal{A}}_t^{n_j}, && (77)
\end{aligned}
$$

and for a general body k_j we have

$$
{}^{0_j}\underline{\bar{\mathcal{A}}}^{k_j} = \underline{\Gamma}^{k_j}\ {}^{0_i}\underline{\bar{\mathcal{A}}}^{p_j} + \underline{\Theta}^{k_j} + \underline{\Xi}^{k_j}\ {}^{0_j}\underline{\mathcal{A}}_t^{n_j}, \tag{78}
$$

where the definition of $\underline{\Gamma}$ is unchanged and $\underline{\Xi}$ follows its original recursion of (53) but is exactly $\underline{\chi}$ on the first body in a new cell.

The $\underline{\Theta}$ term is a known constant which may be propagated forward throughout the coupled closed loops. On the base body $\underline{\Theta}^{0_i} = \underline{0}$ and the recursion is given by

$$
\underline{\Theta}^{k_i} = \underline{\delta}^{k_i}\ \underline{\Theta}^{\Pr[k_i]}, \tag{79}
$$

except for any *first* body which begins a new loop j which is instead given by

$$
\underline{\Theta}^{k_j} = \underline{\delta}^{k_j}\ (\underline{\Theta}^{\Pr[k_j]} + \underline{\Xi}^{\Pr[k_j]}\ {}^{0_j}\underline{\mathcal{A}}_t^{n_i}). \tag{80}
$$

The additional $\underline{\Theta}$ term required for true $O(n+m)$ performance becomes part of the known and applied force term ${}^{0_i}\underline{\hat{\mathcal{F}}}^{n_i}$ in exactly the same manner as the ${}^{0_i}\underline{\mathcal{A}}_t^{n_i}$ terms.

Through appropriate selection of inertia properties, all ladder systems given by Figure 9 can be made to exhibit the same independent coordinate trajectories (q_1 and q_2). Figure 10 shows the independent coordinate trajectories of several such mechanisms (a cell being an independent closed loop) using recursive coordinate reduction (RCR) as well as the solution of an equivalent two degree of freedom spatial pendulum obtained with AUTOLEV.

Figure 11 indicates the performance characteristics for analysis of such a heavily constrained problems: (i) if the system constraints are not considered; (ii) the

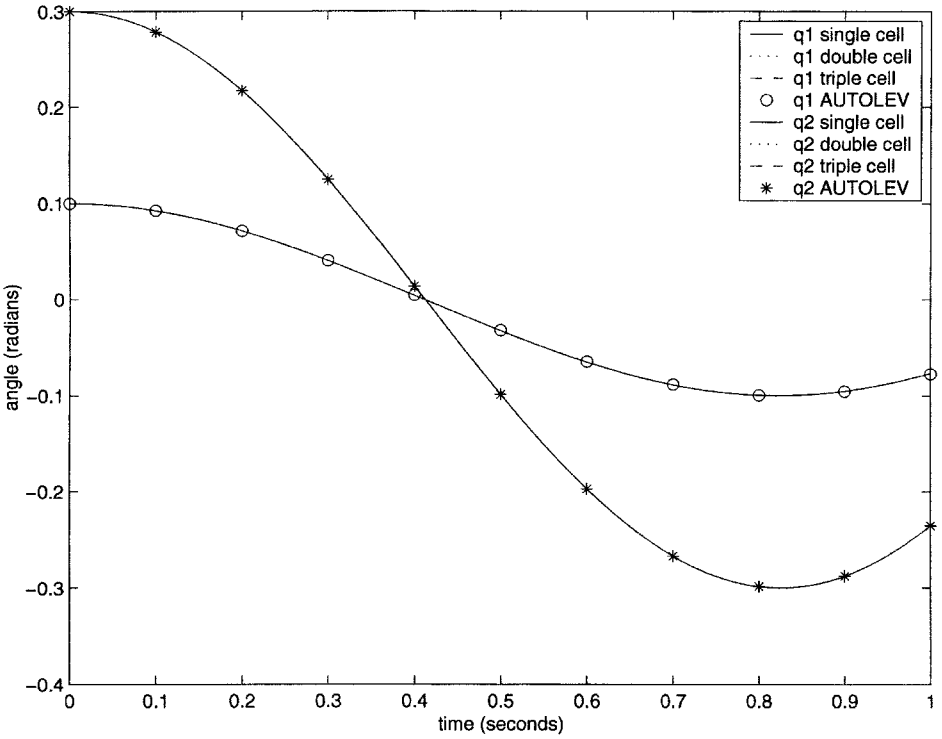


Figure 10. Independent coordinate trajectories of various ladder mechanisms.

system constraints are considered, but the traditional $O(n)$ constraint approach is used; and, (iii) the RCR algorithm presented here is used. Figure 11 indicates that in situations, such as this, where the number of constraints m is of the same order as the number of generalized coordinates n, then even these so-called 'Order-n' algorithms actually offer $O(n^3)$ performance, and generally may not perform even as well as $O(n^3)$-based algorithms. However, the presented recursive coordinate reduction algorithm does not pay such a price, being linear in both the number of generalized coordinates n and the number of algebraic constraints m (e.g. $O(n+m)$).

4.4. RESTRICTIONS

Application of the method of recursive coordinate reduction has been shown to produce valid results for a significant class of problems. However, no discussion is included for systems of coupled loops which do not share a common base body.

The careful observer will also note that multibody systems can be constructed such that the first $n_i - m_i$ joint coordinates in a given loop do not constitute a valid selection of independent coordinates. Moreover, some systems posses singu-

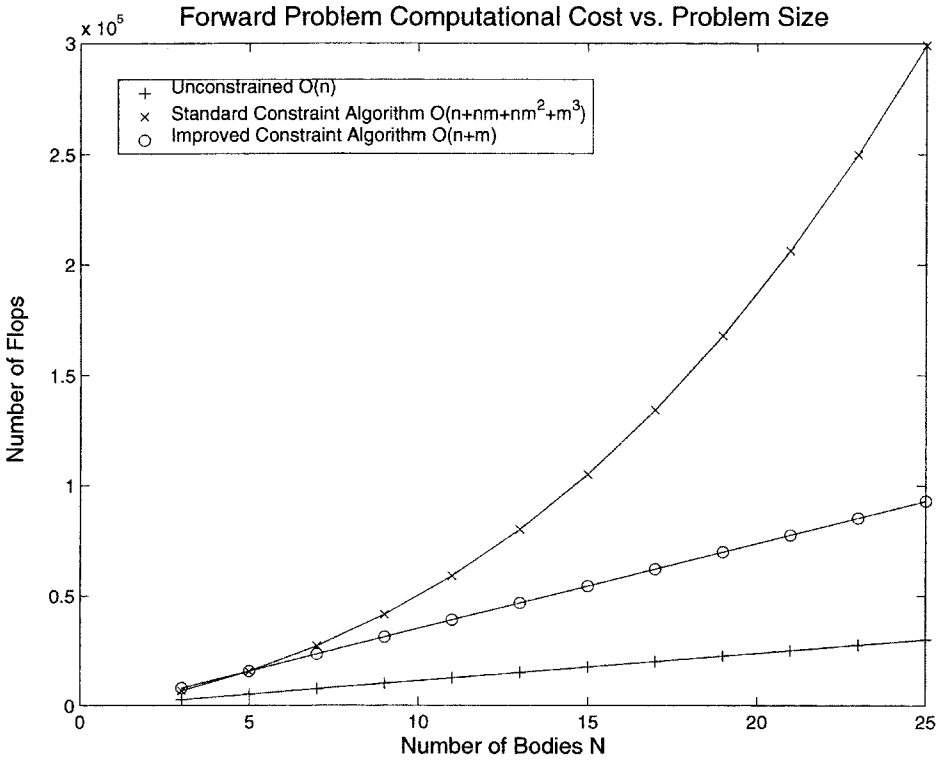


Figure 11. Computational cost required for forward problem simulation of this spatial ladder system.

lar configurations with respect to specific independent coordinate selections at and *near* which a previously valid coordinate selection is invalid or ill conditioned.

These issues are to be addressed in a forthcoming publication.

5. Conclusions

A new method for fully recursive treatment of constraints has been presented in the form of the Recursive Coordinate Reduction algorithm. Relative to the common $O(n)$ constraint method the RCR algorithm exhibits both superior numerical stability of the constraint relations (with respect to temporal integration), and a significant reduction in computational order from $O(n + mn + m^2n + m^3)$ to $O(n + m)$. The method has demonstrated applicability to a large family of closed loop multibody systems, and the extension to the complete set is the topic of a forthcoming publication.

Acknowledgment

Support for this work received under National Science Foundation through award No. 9733684 is gratefully acknowledged.

References

1. Anderson, K.S., 'Recursive derivation of explicit equations of motion for efficient dynamic/control simulation of large multibody systems', Ph.D. Dissertation, Stanford University, 1990.
2. Anderson, K.S., 'An order-N formulation for motion simulation of general constrained multi-rigid-body systems', *Computers & Structures* **43**(3), 1992, 565–572.
3. Anderson, K.S., 'An efficient formulation for the modeling of general multi-flexible-body constrained systems', *International Journal of Solids and Structures* **30**(7), 1993, 921–945.
4. Armstrong, W.W., 'Recursive solution of the equations of motion of an N-link manipulator', in *Proceedings of the Fifth World Congress on the Theory of Machines and Mechanisms*, Montreal, Canada, 1979, Vol. 2, 1342–1346.
5. Bae, D.S. and Haug, E.J., 'A recursive formation for constrained mechanical systems dynamics: Part I, Open loop systems', *Mechanisms, Structures, and Machines* **15**(3), 1987, 359–382.
6. Bae, D.S. and Haug, E.J., 'A recursive formation for constrained mechanical systems dynamics: Part II, Closed loop systems', *Mechanisms, Structures, and Machines*, **15**(4), 1987, 481–506.
7. Banerjee, A.K., 'Block-diagonal equations for multibody elastodynamics with geometric stiffness and constraints', *Journal of Guidance, Control, and Dynamics* **16**(6), 1993, 1092–1100.
8. Baraff, D., 'Linear-time dynamics using Lagrange multipliers', in *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, August 4–9, ASME, New York, 1996, 137–146.
9. Baumgarte, J., 'Stabilization of constraint and integrals of motion in dynamical systems', *Computer Methods in Applied Mechanics and Engineering* **1**, 1972, 1–16.
10. Brenan, K.E., Campbell, S.L. and Petzold, L.R., *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, North-Holland, New York, 1989.
11. Chace, M.A., 'Method and experience in computer aided design of large-displacement mechanical systems', in *Computer Aided Analysis and Optimizations of Mechanical System Dynamics*, E.J. Haug (ed.), Springer-Verlag, Berlin, 1984, 233–259.
12. Featherstone, R., 'The calculation of robotic dynamics using articulated body inertias', *International Journal of Robotics Research* **2**(1), 1983, 13–30.
13. Featherstone, R., *Robotic Dynamics Algorithms*, Kluwer Academic Publishers, Dordrecht, 1987.
14. Garcia de Jalon, J. and Bayo, E., *Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge*, Springer-Verlag, New York, 1994.
15. Hollerbach, J.M., 'A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity,' *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-10**(11), 1980, 730–736.
16. Jain, A. and Rodriguez, G., 'Linearization of manipulator dynamics using spatial operators', *IEEE Transactions on Systems, Man and Cybernetics*, **23**(1), 1993, 239–248.
17. Jain, A. and Rodriguez, G., 'Diagonalized Lagrangian robot dynamics', *IEEE Transactions on Robotics and Automation* **11**(4), 1995, 571–584.
18. Kane, T.R. and Levinson, D.A., *Dynamics: Theory and Application*, McGraw Hill, New York, 1985.
19. Lubich, C., Nowak, U., Pohle, U. and Engstler, C., 'MEXX-numerical software for the integration of constrained mechanical systems', Berlin, SC92-12, 1992.

20. Nielan, P.E., 'Efficient computer simulation of motions of multibody systems', Ph.D. Dissertation, Stanford University, 1986.
21. Orlandea, N., Chace, M.A. and Calahan, D.A., 'A sparsity-oriented approach to the dynamics analysis and design of mechanical systems – Parts 1 & 2', *Journal of Engineering for Industry* **43**, 1977, 773–784.
22. Park, K.C. and Chiou, J.C., 'Stabilization of computational procedures for constrained dynamical systems', *Journal of Guidance, Control and Dynamics* **11**(4), 1988, 365–370.
23. Petzold, L.R., 'Differential/algebraic equations are not ODE's', *SIAM Journal of Science, Statistics and Computation* **3**(3), 1982, 367–384.
24. Petzold, L.R. and Lotstedt, P., 'Numerical solution of nonlinear differential equations with algebraic constraints II: Practical implications', *SIAM Journal of Science, Statistics and Computation* **7**(3), 1986, 720–733.
25. Rosenthal, D.E. and Sherman, M.A., 'High performance multibody simulations via symbolic equation manipulation and Kane's method', *Journal of the Astronautical Sciences* **34**(3), 1986, 223–239.
26. Rosenthal, D.E., 'An order n formulation for robotic systems', *Journal of Astronautical Sciences* **38**(4), 1990, 511–529.
27. Saha, S.K. and Schiehlen, W.O., 'Recursive kinematics and dynamics for parallel structured closed-loop multibody systems', *Mechanical Structures and Machines* **29**(2), 2001, 143–175.
28. Schwertassek, R., 'Reduction of multibody simulation time by appropriate formulation of dynamical system equations', in *Proceedings of the NATO Advanced Study Institute on Computer-Aided Analysis of Rigid and Flexible Mechanical Systems*, Troia, Portugal, June 27–July 9, M.S. Pereira and J.A.C. Ambrósio (eds.) Kluwer Academic Publishers, Dordrecht, 1993, 447–482.
29. Stajskal, V. and Valášek, M., *Kinematics and Dynamics of Machinery*, Marcel Dekker, New York, 1996.
30. Vereshchagin, A.F., 'Gauss principle of least constraint for modeling the dynamics of automatic manipulators using a digital computer', *Soviet Physics – Doklady* **20**(1), 1975, 33–34.
31. Walker, M.W. and Orin, D.E., 'Efficient dynamic computer simulation of robotic mechanisms', *Journal of Dynamic Systems, Measurements, and Control* **104**, 1982, 205–211.
32. Wehage, R.A. and Haug, E.J., 'Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems', *Journal of Mechanical Design* **104**, 1982, 247–255.
33. Wehage, R.A., 'Application of matrix partitioning and recursive projection to $O(n)$ solution of constrained equations of motion', in *ASME Design Engineering Division (Publication) DE*, Vol. 15-2, Trends and Developments in Mechanisms, ASME, New York, 1988, 221–230.