To appear in journal Nonlinear Dynamics 2006

Orthogonal Complement Based Divide-and-Conquer Algorithm (O-DCA) for Constrained Multibody Systems

Rudranarayan M. Mukherjee, Kurt S. Anderson
Computational Dynamics Laboratory
Department of Mechanical Aerospace and Nuclear Engineering.
Rensselaer Polytechnic Institute
110 8th Street. Troy NY 12180 USA
e-mail: mukher@rpi.edu, anderk5@rpi.edu

March 15, 2006

Abstract

Keywords: Closed Kinematic Loops, Orthogonal Complement, Singular Configurations, Logarithmic Computational Complexity, Divide and Conquer.
A new algorithm is presented in this paper for calculating the forward dynamics of multi-rigid bodies connected together by kinematic joints to form single or coupled closed loop topologies. The algorithm is exact and non-iterative. The constraints are imposed at the acceleration level by utilizing a kinematic relation between the joint motion subspace (or partial velocities) and its orthogonal complement. Sample test cases indicate excellent constraint satisfaction and robust handling of singular configurations. Since the present algorithm does not use either a reduction or augmentation approach in the traditional sense for imposing the constraints, it does not suffer from the associated problems for systems passing through singular configurations. The computational complexity of the algorithm is expected to be $O(n+m)$ and $O(log(n+m))$ for serial and parallel implementation respectively, where *n* is the number of generalized coordinates and *m* is the number of independent algebraic constraints.

1 Introduction

Computer simulation and associated analysis of the dynamic behavior of multibody systems is an essential tool for engineers and researchers working in various fields. The involved applications include, but are not limited to, terrestrial and space vehicles, bio-mechanical systems, materials modelling, robotics and manufacturing processes. For such model-based engineering to be effective, it is essential that the simulation tools used be computationally efficient, accurate, and robust. Thus, the development of algorithms to model multibody system dynamics has been an active area of research.

Several algorithms of various computational complexities have been presented in the literature. The earliest algorithms for articulated body systems were of $O(n^3)$ complexity [1] (the number of computational operations increase as a cubic function of n, per integration step), with n being the number of degrees of freedom of the system. In the late 1970s through the early 1990s emphasis was placed by a number of researchers on the development of lower computational order (cost) algorithms [2]-[5]. Several algorithms were independently derived and developed by various authors for solving the multi-rigid body and multi-flexible dynamics problem in $O(n)$ complexity [6]-[13]. A brief review and the underlying similarities between many of these different algorithms is discussed in reference [14].

Many multibody systems of scientific and engineering interest include topologies with closed kinematical loops. In these situations, the closed kinematical loops are most often modelled by producing a set of n equations of motion associated with the unconstrained system, and a companion set of m independent algebraic constraint equations which must be satisfied throughout the solution of the equations of motion. These constraint equations may then be used to either: *i)* Reduce out excessive degrees of freedom producing a

$\vec{r}$	Position vector
$\vec{\omega}^i$	Absolute angular velocity of reference frame *i* on body *i*, a 3 × 1 matrix
$\vec{v}^i$	Absolute translational velocity of a point *i*, a 3 × 1 matrix
$\vec{\alpha}^i$	Absolute angular acceleration of reference frame *i* on body *i*, a 3 × 1 matrix
$\vec{a}^i$	Absolute translational acceleration of a point *i*, a 3 × 1 matrix
V^i	Spatial velocity of point *i* associated with frame *i* = $\begin{bmatrix}\vec{\omega}^i\\ \vec{v}^i\end{bmatrix}_{6\times1}$
A^i	Spatial acceleration of point *i* associated with frame *i* = $\begin{bmatrix}\vec{\alpha}^i\\ \vec{a}^i\end{bmatrix}_{6\times1}$
$\boldsymbol{H}^{k/(k+1)}$	Joint motion map of the kinematic joint between bodies k and k+1
$\boldsymbol{D}^{k/(k+1)}$	Orthogonal Complement of the kinematic joint between bodies k and k+1
J^{k+}	Joint connecting bodies k-1 and k
J^{k-}	Joint connecting bodies k and k+1
q	Generalized relative coordinate
u	Generalized relative speed
$\dot{u}$	Time derivative of generalized relative speed
$\boldsymbol{\zeta}$	Inertia coupling terms for individual body or subassembly
$\boldsymbol{\Upsilon}$	Inertia coupling terms for assembly
$\vec{\tau}_c^{\,i}$	Constraint torque at joint *i*, 3×1 column matrix
$\vec{f}_c^{\,i}$	Constraint force at joint *i*, 3×1 column matrix
$\tilde{\tau}_c^{\,i}$	Measure numbers of constraint torque at joint *i*, i.e. ordered list of non-zero elements of $\vec{\tau}_c^{\,i}$
$\tilde{f}_c^{\,i}$	Measure numbers of constraint force at joint *i*, i.e. ordered list of non-zero elements of $\vec{f}_c^{\,i}$
$F_c^{\,i}$	Spatial constraint force at joint *i* = $\begin{bmatrix}\vec{\tau}_c^{\,i}\\ \vec{f}_c^{\,i}\end{bmatrix}_{6\times1}$
$\tilde{F}_c^{\,i}$	Measure numbers i.e ordered list of non-zero elements of $F_c^{\,i}$ = $\begin{bmatrix}\vec{\tau}_c^{\,i}\\ \vec{f}_c^{\,i}\end{bmatrix}_{(6-dof)\times1}$
$\vec{b}\times$	3×3 skew symmetric matrix for cross product of any vector $\vec{b}$
n	The number of generalized coordinates in the system
m	The number of independent algebraic constraints
bb and tb	Base and Terminal joints through which the four bar linkage shown in Figure (3) connects to ground
sys	Superscript representing the whole system as a single body
asm	Superscript representing the bodies 2,4,5 and 6 from Figure(4) as a single body
C^T	Transpose of any arbitrary matrix C
$\boldsymbol{U}$	Identity matrix
$\boldsymbol{Z}$	Zero matrix
$\widehat{\boldsymbol{W}}$	Useful Intermediate Quantity
$\widehat{\boldsymbol{X}}$	Useful Intermediate Quantity
$\widehat{\boldsymbol{Y}}$	Useful Intermediate Quantity
$\boldsymbol{X}$	Useful Intermediate Quantity
$\boldsymbol{Y}$	Useful Intermediate Quantity
$\boldsymbol{K}$	Useful Intermediate Quantity
$\boldsymbol{D}$	Useful Intermediate Quantity
$*$	All bold faced symbols and letters represent matrix quantities

Table 1: The Nomenclature

minimum dimension system of $n - m$ equations; or *ii)* Augment the equations of motion producing a larger $n + m$ dimension system of equations involving m redundant state variables. The first $O(n)$ methods for

systems with closed kinematical loops were independently presented in references [15][16] and were of the augmentation category, while coordinate partitioning [17] and Recursive Coordinate Reduction [18][19] were of the reduction type.

Whichever type of procedure is used, there are two principal problems encountered when dealing with systems with closed kinematical loops, viz. (1) the saddle point problem originating from constraint equations becoming linearly dependent and (2) the accumulation of integration errors leading to significant drift in constraint satisfaction. The saddle point problem is typically encountered when the system passes through a singular configuration whereby the constraint Jacobian becomes rank deficient. A related problem may additionally occur for reduction approaches when the *dependency* matrix relating the dependent state derivative(s) to the independent state derivative(s), which is necessary for such a formulation, loses rank. By comparison, the problem of constraint violation error drift can be traced back to the fact that unless some form of constraint stabilization approach is used, the constraint are most often imposed at the accelerations, or possibly the velocity level. Imposing the constraints at the acceleration level results in an eventual unstable growth in constraint violation which can grow exponentially for a given simulation. The error in constraint violation occurs due to the accumulation of round-off errors from finite precision arithmetic and the introduction of two zero eigenvalues (one associated with each constant of temporal integration) for each acceleration level constraint. To overcome this problem some form of constraint stabilization is often introduced into the equations. Constraint stabilization techniques have been used in various forms for a number of years. The most common of these can be found in reference [20]-[26]. Although introducing a constraint stabilization technique can reduce the drift in the constraint violation significantly, these methods do not generally provide full constraint satisfaction, and come with their own (potentially significant) computational cost. Unfortunately, in many situations involving stiff systems and/or systems repeatedly passing near and through singular configurations, the constraint violation errors can grow rapidly and result in a significant loss of accuracy, making some form of stabilization essential.

Another concern which may arise when dealing with complex systems is the considerable additional expense incurred when dealing with heavily constrained systems ($m \sim n$). With so-called $O(n)$ complexity algorithms, the simulation turn around times scale linearly with the increase in system size (number of generalized coordinates n) and hence are more efficient than the traditional $O(n^3)$ approaches when dealing with articulated body systems where $n \gg 1$. Unfortunately, these algorithms do not perform quite as well when one is dealing with systems involving many kinematic loops. In such instances these so-called $O(n)$ algorithms actual perform as $O(n + nm^2 + m^3)$. Other strategies exist [18][19][27][28] which offer $O(n + m)$ overall performance, but these procedures are significantly less easy to implement and are not as a rule applicable to all system topologies.

If one wishes to exploit the potential advantages in reduced simulation turnaround time through the use parallel computing, then the theoretical lower limit on effective cost per integration step (turn around time per temporal integration step) is $O(log(n))$. Thus, if the computations are sufficiently coarse grain parallel and inter-processor communications costs (e.g. communications latency and data transfer) are adequately low, then substantial gains may be potentially realized through the use of parallel computing.

The first parallel algorithm which was both *time optimal* $O(log(N))$ turn around time per temporal integration step) and *processor optimal* (theoretically achieves this $O(log(N))$ turn around with only $O(N)$ processors) was presented in [29], but was limited to chain systems of N bodies. In [30][31] a Divide and Conquer Algorithm is presented that can achieve $O(log(n))$ complexity when implemented on $O(N)$ processors in parallel and is applicable for general topologies. The extension of the algorithm for systems with closed kinematical loops uses a constraint stabilization method together with a formulation utilizing Lagrange multipliers. The method can degenerate to $O(n^3)$ complexity (if solved sequentially) in the worst case. Moreover a necessary matrix used for dealing with loops within the procedure can become rank deficient and in such cases the method requires an alternate formulation.

In this paper, an algorithm is proposed for handling systems with closed kinematical loops. The proposed method uses a Divide and Conquer formulation similar to that in [30]. This formulation is time and processor optimal, but does not include coordinate reduction or Lagrange multipliers for constraint imposition. The procedure implements the spatial Newton-Euler formulation in a Divide and Conquer scheme and imposes the constraints at the acceleration level by using a kinematic relationship involving the orthogonal complement of the joint motion subspace. The method proposed can easily handle systems in truly singular configurations, and is applicable for general systems containing either single closed loops or multiple coupled closed loops.

2 Analytical Preliminaries

This section presents a brief review of the Divide and Conquer algorithm for serial chains as found in [30]. This background is essential to understand the handling of non-terminal bodies in the closed loop topologies because this treatment is effectively similar to the method presented in [30] for chain systems. In section (3) the new procedure is presented for handling loop closure constraints at the terminal bodies.

The basic unit of the DCA scheme is the two-handle representation of a body. A handle is any selected point on the body which is used in modelling the interactions of the body with the environment. The handles on a body can correspond to a joint location, a center of mass or any desired reference point. The two handles can even coincide. A body can have any number of handles on it. For the algorithm presented here, the joint locations are chosen as the handles on the body.

Consider two representative bodies *Body* k and *Body* $k+1$ of the articulated body system as shown in Figure (1). The two handles on *Body* k correspond to the joints J^{k+} and J^{k^-}. Similarly, the two handles on *Body* $k+1$ correspond to the joints J^{k+1^+} and J^{k+1^-}.

Using a spatial Newton-Euler formulation, the equations of motion of a representative body *Body* k of the system can be written at the two handles as below

$$A^{k^+} = \zeta^k_{11} F^{k+}_c + \zeta^k_{12} F^{k^-}_c + \zeta^k_{13} \quad (1)$$

$$A^{k^-} = \zeta^k_{21} F^{k+}_c + \zeta^k_{22} F^{k^-}_c + \zeta^k_{23} \quad (2)$$

Here A^{k^+} and A^{k^-} are the spatial accelerations of the body at the handles at J^k+ and J^k-, respectively. The terms ζ^k_{ij} $(i = 1, 2\ j = 1, 2)$ are the inverse of the spatial inertia terms while the terms ζ^k_{i3} $(i = 1, 2)$ are the inertia dependent bias terms. The bias terms also contain contributions from any forces applied to the body which are determinable directly from the system state. These active forces include body forces, actuator forces, spring-damper forces etc. The terms F^{k+}_c and $F^{k^-}_c$ are the unknown constraint forces acting on the body at the joint locations. The interactions of the body with the rest of the system are achieved through these constraint forces. At the beginning of the simulation, the inertia dependent terms viz. ζ^k_{ij} as well as the active forces for each body are either known or can be easily calculated from the state of the system. The above equations then reduce to two sets of equations in two sets of unknowns viz. the spatial accelerations (A^{k^+} , A^{k^-}), and the constraint forces (F^{k+}_c , $F^{k^-}_c$). This set of equations are henceforth referred to as the two handle equations of motion of a *Body* k. Similarly the two handle equations of motion for *Body k+1* can be written in the form

$$A^{k+1^+} = \zeta^{k+1}_{11} F^{k+1^+}_c + \zeta^{k+1}_{12} F^{k+1^-}_c + \zeta^{k+1}_{13} \quad (3)$$

$$A^{k+1^-} = \zeta^{k+1}_{21} F^{k+1^+}_c + \zeta^{k+1}_{22} F^{k+1^-}_c + \zeta^{k+1}_{23} \quad (4)$$

There are two main processes in the DCA approach, a hierarchic assembly process and a hierarchic disassembly process. In the hierarchic assembly process, the equations of motion of each body are written in terms of the accelerations at each of its two handles (the procedure may be easily generalized to bodies with more than two handles). The two handle equations of motion of two successive bodies are then coupled together to form the two handle equations of motion of the resulting assembly using a recursive set of formulate derived in section (2.1)

$$A^{k^+} = \boldsymbol{\Upsilon}_{11} F^{k+}_c + \boldsymbol{\Upsilon}_{12} F^{k+1^-}_c + \boldsymbol{\Upsilon}_{13} \quad (5)$$

$$A^{k+1^-} = \boldsymbol{\Upsilon}_{21} F^{k+}_c + \boldsymbol{\Upsilon}_{22} F^{k+1^-}_c + \boldsymbol{\Upsilon}_{23} \quad (6)$$

The two handles of the resulting assembly are the inward joint of the *Body k* (viz. J^{k+}) and the outward joint on the *Body k+1* (viz. J^{k+1^-}) and the constraint forces are those acting on the resulting assembly at those handles. The inertia coupling terms, Υ_{ij}, for the resulting assembly are calculated using a recursive set of formulae as discussed in section (2.1) of this paper.

This process begins at the level of individual bodies of the system. Adjacent bodies of the system are hierarchically assembled to construct a binary tree as shown in Figure (2). Individual bodies that make up the system form the leaf (base) nodes of the binary tree. The equations of motion of a pair of bodies are coupled

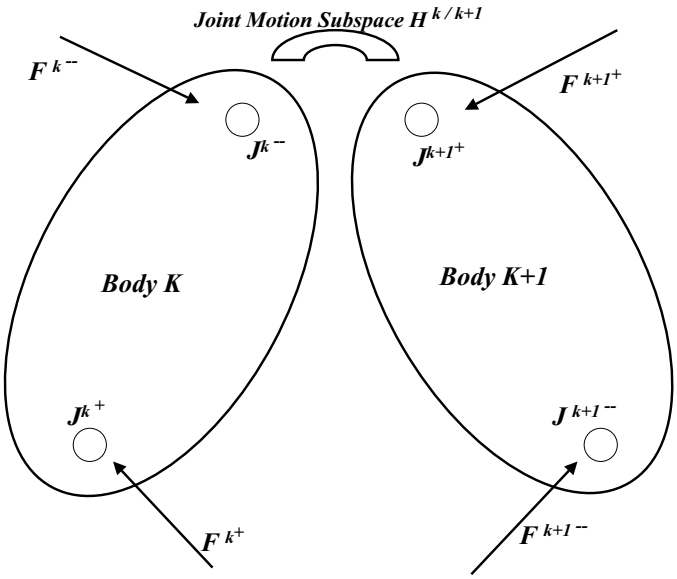


Figure 1: Two Handle Articulated Body

together using the recursive set of formulae to form the two handle equations of motion of the resulting assembly. The resulting assembly now corresponds to a node of the next level in the binary tree. Working up the binary tree in this hierarchic assembly processes, only a single assembly is left as the root (top) node of the binary tree. The root (top) node corresponds to the two-handle representation of the entire articulated system modelled as a single assembly. The two handles on this body correspond the boundary joints of the articulated system.

The procedure for solving the equations of motion of the top node using the boundary conditions is discussed in detail for systems with kinematically closed loops in section (3). Using the procedure outlined in there, the spatial accelerations and the constraint forces on the top node at the terminal handles can be generated. The hierarchic disassembly process begins with the solution of the two-handle equations of motion of the top node. From this solution, the spatial accelerations of and forces on the two handles of the single assembly are known. The spatial acceleration and constraint forces generated by solving the two handle equations of an assembly are identically the values of the spatial accelerations and constraint forces on one handle on each of the two constituent assemblies. From these known quantities, the two handle equations of motion of the constituent assemblies can be easily solved to obtain the constraint force and spatial acceleration at the connecting joint. For example, for a representative assembly made from *Body k* and *Body k+1*, the equations of motion are given by equations (5-6). On solving these equations the quantities A^{k^+}, A^{k+1^-}, $F_c^{k^+}$ and $\mathrm{F_c^{k+1^-}}$ are generated. These quantities are then substituted into the two-handle equations of the constituent sub-assemblies say for *Body k* and *Body k+1*. Thus knowing the values of A^{k^+}, $F_c^{k^+}$, equations (1-2) can be easily solved, while from A^{k+1^-} and $F_c^{k+1^-}$ equations (3-4) can be solved. This process is repeated in a hierarchic disassembly of the binary tree where the known boundary conditions are used to solve the two-handle equations of motion of the immediate subassemblies, until spatial acceleration and constraint forces on all bodies in the system are calculated.

Similar to the scheme in [30], this algorithm works in four sweeps, traversing the system topology like a binary tree. The first and the third sweep work upwards from the leaf (base) nodes of the binary tree to the top node while the second and the fourth sweep work downwards. The input to this algorithm is comprised of the mass properties of the bodies, joint generalized coordinates and speeds. The first two sweeps generate

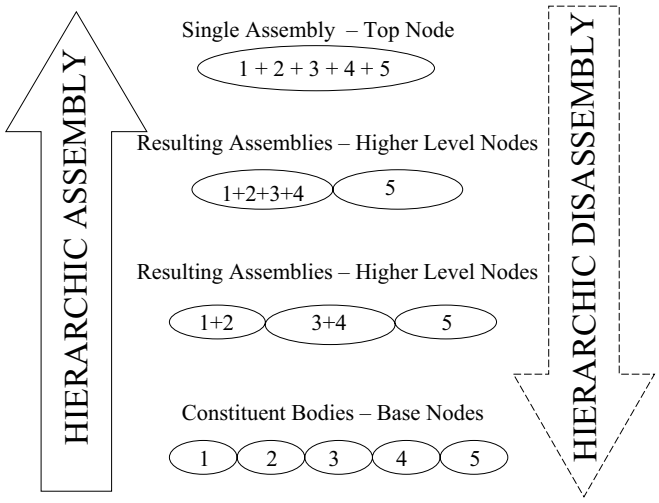


Figure 2: The Hierarchic Assembly and Disassembly Process using Binary Tree Structure

the position and velocity of each handle on each node by using an assembly-disassembly process similar to that described in [30]. On completing the two sweeps, on each base node, the coordinate transformations, the state dependent accelerations, the active joint forces are calculated. The active forces are state dependent and include actuator forces on the joints, damping forces and body forces like gravity. The final two sweeps correspond to the hierarchic assembly and the hierarchic disassembly processes respectively.

In the analytical treatment presented here, direction cosine matrices and transformation between different basis are not shown explicitly. Appropriate basis transformations have to be taken into account for an implementation of this algorithm. Also, this algorithm uses a redundant mixed set of coordinates, viz. Cartesian coordinates and relative coordinates, throughout the derivation. Mixed set of coordinates has been used in [32]-[33] for rigid body dynamics.

2.1 Recursive Formulae

The relative acceleration at the joint connecting *Body* k and *Body* $k+1$ is given by the following equation

$$ {}^{N}A^{k+1^{+}} - {}^{N}A^{k^{-}} = \boldsymbol{H}^{k/(k+1)}\dot{u} + \dot{\boldsymbol{H}}^{k/(k+1)}u \tag{7} $$

In this equation, $\boldsymbol{H}^{k/(k+1)}$ is the joint motion subspace matrix, u and $\dot{u}$ are the relative generalized speeds, and relative generalized accelerations (translation and/or rotation), respectively, at the joint. The column vector(s) of the joint motion subspace matrix are the basis of the space which contains the active forces at the joint and the joint degrees of freedom (dof). It can be interpreted as the $6 \times dof$ matrix that maps the dof generalized speeds at the joint into a 6×1 column matrix of spatial relative velocity at the joint. Matrix $\boldsymbol{D}^{k/(k+1)}$ is defined to be the orthogonal complement of the joint motion subspace matrix $\boldsymbol{H}^{k/(k+1)}$. While $\boldsymbol{H}^{k/(k+1)}$ is a $6 \times dof$ matrix corresponding to the $dof \times 1$ column matrix of joint degrees of freedom, $\boldsymbol{D}^{k/(k+1)}$ is a $6 \times (6 - dof)$ matrix that maps the constrained degrees of freedom of the joint. The column vector(s) of $\boldsymbol{D}^{k/(k+1)}$ are the basis of the space in which the constraint forces acting on the joint lie (i.e. the column matrices are the basis of this space where the joint can support constraint forces). For example, in a spherical joint, the translational degrees of motion are constrained while the rotational degrees of freedom

are maintained. Hence the corresponding maps maybe given by

$$\boldsymbol{H}^{k/(k+1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{D}^{k/(k+1)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

By definition of the orthogonal complement $\boldsymbol{H}^{k/(k+1)}$ and $\boldsymbol{D}^{k/(k+1)}$ satisfy the following relation

$$\boldsymbol{H}^{k/(k+1)^T} \cdot \boldsymbol{D}^{k/(k+1)} = \boldsymbol{D}^{k/(k+1)^T} \cdot \boldsymbol{H}^{k/(k+1)} = 0 \tag{9}$$

Newton's Third Law requires the constraint force on joint J^{k+1^+} viz. $F_c^{k+1^+}$ and joint J^{k^-} viz. $F_c^{k^-}$ are equal in magnitude and opposite in direction. Using this relation and substituting the expressions for ${}^N A^{k^-}$ and ${}^N A^{k+1^+}$ from equation (2) into equation (3) into equation (7), an expression for $F_c^{k+1^+}$ is obtained as

$$\begin{aligned}[\zeta_{11}^{k+1} + \zeta_{22}^{k}] F_c^{k+1^+} = [\zeta_{21}^{k} F_c^{k+} - \zeta_{12}^{k+1} F_c^{k+1^-} + \zeta_{23}^{k} \\ -\zeta_{13}{}^{k+1} + \boldsymbol{H}^{k/(k+1)} \dot{u} + \dot{\boldsymbol{H}}^{k/(k+1)} u]\end{aligned} \tag{10}$$

$$\begin{aligned}\Rightarrow F_c^{k+1^+} = [\zeta_{11}^{k+1} + \zeta_{22}^{k}]^{-1} [\zeta_{21}^{k} F_c^{k+} - \zeta_{12}^{k+1} F_c^{k+1^-} \\ +\zeta_{23}^{k} - \zeta_{13}^{k+1} + \boldsymbol{H}^{k/(k+1)} \dot{u} + \dot{\boldsymbol{H}}^{k/(k+1)} u]\end{aligned} \tag{11}$$

Premultiplying equation (10) by $\boldsymbol{D}^{k/(k+1)^T}$ gives

$$\begin{aligned}\boldsymbol{D}^{k/(k+1)^T} [\zeta_{11}^{k+1} + \zeta_{22}^{k}] F_c^{k+1^+} = \boldsymbol{D}^{k/(k+1)^T} [\zeta_{21}^{k} F_c^{k+} + \zeta_{23}^{k} - \zeta_{13}^{k+1} - \zeta_{12}^{k+1} F_c^{k+1^-} \\ +\dot{\boldsymbol{H}}^{k/(k+1)} u] + \underbrace{\boldsymbol{D}^{k/(k+1)^T} \boldsymbol{H}^{k/(k+1)}}_{0} \dot{u}\end{aligned} \tag{12}$$

From the definition of the orthogonal complement of joint motion subspace, the constraint force $F_c^{k+1^+}$ can be expressed in terms of the measure numbers of the constraint torques and constraint forces as

$$F_c^{k+1^+} = \boldsymbol{D}^{k/(k+1)} \tilde{F}^{k+1^+} \tag{13}$$

where the constraint force and constraint moment measure numbers $\tilde{f}_c{}^{k+1^+}$ and $\tilde{\tau}_c{}^{k+1^+}$, respectively, are represented as

$$\tilde{F}^{k+1^+} = \begin{bmatrix} \tilde{\tau}_c{}^{k+1^+} \\ \tilde{f}_c{}^{k+1^+} \end{bmatrix} \tag{14}$$

Substituting relation (13) into equation (12) yields

$$\begin{aligned}\boldsymbol{D}^{k/(k+1)^T} [\zeta_{11}^{k+1} + \zeta_{22}^{k}] \boldsymbol{D}^{k/(k+1)} \tilde{F}^{k+1^+} = \boldsymbol{D}^{k/(k+1)^T} [\zeta_{21}^{k} F_c^{k+} - \zeta_{12}^{k+1} F_c^{k+1^-} \\ +\zeta_{23}^{k} - \zeta_{13}^{k+1} + \dot{\boldsymbol{H}}^{k/(k+1)} u])\end{aligned} \tag{15}$$

The term $\boldsymbol{D}^{k/(k+1)^T} [\zeta_{11}^{k+1} + \zeta_{22}^{k}] \boldsymbol{D}^{k/(k+1)}$ appearing in (15) is a Symmetric Positive Definite (SPD) matrix and hence there is no problem associated with its inversion. Defining the quantity $\widehat{X}$ as

$$\widehat{\boldsymbol{X}} \triangleq \boldsymbol{D}^{k/(k+1)^T} [\zeta_{11}^{k+1} + \zeta_{22}^{k}] \boldsymbol{D}^{k/(k+1)} \tag{16}$$

$\tilde{F}^{k+1^+}$ may be determined as

$$\tilde{F}^{k+1^+} = \widehat{\boldsymbol{X}}^{-1}\boldsymbol{D}^{k/(k+1)^T}[\boldsymbol{\zeta}_{21}^k F_c^{k+} - \boldsymbol{\zeta}_{12}^{k+1}F_c^{k+1^-} + \boldsymbol{\zeta}_{23}^k - \boldsymbol{\zeta}_{13}^{k+1} + \dot{\boldsymbol{H}}^{k/(k+1)}u]. \tag{17}$$

The above expression (17) is then premultiplied by $\boldsymbol{D}^{k/(k+1)}$ to get the desired expression for the spatial constraint force $F_c^{k+1^+}$,

$$\begin{aligned} F_c^{k+1^+} &= \boldsymbol{D}^{k/(k+1)}\tilde{F}^{k+1^+} \\ &= \boldsymbol{D}^{k/(k+1)}\widehat{\boldsymbol{X}}^{-1}\boldsymbol{D}^{k/(k+1)^T}[\boldsymbol{\zeta}_{21}^k F_c^{k+} - \boldsymbol{\zeta}_{12}^{k+1}F_c^{k+1^-} + \boldsymbol{\zeta}_{23}^k - \boldsymbol{\zeta}_{13}^{k+1} + \dot{\boldsymbol{H}}^{k/(k+1)}u] \end{aligned} \tag{18}$$

The above expression for $F_c^{k+1^+}$ can be compactly written as below

$$F_c^{k+1^+} = \widehat{\boldsymbol{W}}\boldsymbol{\zeta}_{21}^k F_c^{k+} - \widehat{\boldsymbol{W}}\boldsymbol{\zeta}_{12}^{k+1}F_c^{k+1^-} + \widehat{\boldsymbol{Y}} \tag{19}$$

$$\text{where} \quad \widehat{\boldsymbol{W}} = \boldsymbol{D}^{k/(k+1)}\widehat{X}^{-1}\boldsymbol{D}^{k/(k+1)^T} \tag{20}$$

$$\text{and} \quad \widehat{\boldsymbol{Y}} = \widehat{\boldsymbol{W}}[\boldsymbol{\zeta}_{23}^k - \boldsymbol{\zeta}_{13}^{k+1} + \dot{\boldsymbol{H}}^{k/(k+1)}u] \tag{21}$$

This expression for $F_c^{k+1^+}$ is substituted in equations (1) and (4), and after some algebraic manipulation, the two equations can be obtained as

$$A^{k+} = [\boldsymbol{\zeta}_{11}^k - \boldsymbol{\zeta}_{12}^k\widehat{\boldsymbol{W}}\boldsymbol{\zeta}_{21}^k]F_c^{k+} + \boldsymbol{\zeta}_{12}^k\widehat{\boldsymbol{W}}\boldsymbol{\zeta}_{12}^{k+1}F_c^{k+1^-} + \boldsymbol{\zeta}_{13}^k - \boldsymbol{\zeta}_{12}^k\widehat{\boldsymbol{Y}} \tag{22}$$

and

$$A^{k+1^-} = \boldsymbol{\zeta}_{21}^{k+1}\widehat{\boldsymbol{W}}\boldsymbol{\zeta}_{21}^k F_c^{k+} + [\boldsymbol{\zeta}_{22}^{k+1} - \boldsymbol{\zeta}_{21}^{k+1}\widehat{\boldsymbol{W}}\boldsymbol{\zeta}_{12}^{k+1}]F_c^{k+1^-} + \boldsymbol{\zeta}_{23}^{k+1} + \boldsymbol{\zeta}_{21}^{k+1}\widehat{\boldsymbol{Y}} \tag{23}$$

Equations (22) and (23) can be considered as the two handle equations of motion of the resulting assembly of *Body* k and *Body* $k+1$. The two handles on this assembly are the joints J^{k+} and J^{k+1^-}. Collecting terms in above equations, the two handle equations of motion of the assembly can be written as

$$A^{k^+} = \Upsilon_{11}F_c^{k+} + \Upsilon_{12}F_c^{k+1^-} + \Upsilon_{13} \tag{24}$$

$$A^{k+1^-} = \Upsilon_{21}F_c^{k^+} + \Upsilon_{22}F_c^{k+1^-} + \Upsilon_{23} \tag{25}$$

where now Υ_{ij} are the composite inertia of the assembly. From the above, a recursive set of formulae for Υ_{ij} can be obtained as

$$\boldsymbol{\Upsilon}_{11} = [\boldsymbol{\zeta}_{11}^k - \boldsymbol{\zeta}_{12}^k\widehat{\boldsymbol{W}}\boldsymbol{\zeta}_{21}^k] \tag{26}$$

$$\boldsymbol{\Upsilon}_{22} = [\boldsymbol{\zeta}_{22}^{k+1} - \boldsymbol{\zeta}_{21}^{k+1}\widehat{\boldsymbol{W}}\boldsymbol{\zeta}_{12}^{k+1}] \tag{27}$$

$$\boldsymbol{\Upsilon}_{12} = \boldsymbol{\Upsilon}_{21}{}^T = \boldsymbol{\zeta}_{12}^k\widehat{\boldsymbol{W}}\boldsymbol{\zeta}_{12}^{k+1} \tag{28}$$

$$\boldsymbol{\Upsilon}_{13} = \boldsymbol{\zeta}_{13}^k - \boldsymbol{\zeta}_{12}^k\widehat{\boldsymbol{Y}} \tag{29}$$

$$\boldsymbol{\Upsilon}_{23} = \boldsymbol{\zeta}_{23}^{k+1} + \boldsymbol{\zeta}_{21}^{k+1}\widehat{\boldsymbol{Y}} \tag{30}$$

In the associated manipulations, the two bodies are coupled together to form an assembly by expressing the intermediate (common) joint constraint force in terms of the constraint forces at the other two handles. This process can now be repeated for all bodies in the system where the two handle equations of motion of two successive bodies or assemblies are coupled together using the recursive formulae to obtain the two handle equations of the resulting assembly. This process works hierarchically exploiting the same structure as that of a binary tree. At the end of the hierarchic assembly process, the whole articulated system may be modelled in terms of the two handle equations of motion of a single assembly. The methodology outlined here is effectively identical to the procedure outlined in [30], though some intermediate manipulations may appear to be different. The primary import of this section is that an articulated chain system can be modelled as a single assembly with handles at the base and terminal joints of the system. In the next section, a new methodology is outlined that explains how the two handle equations of motion of the resulting assembly can be solved when the base and terminal joints are such that the system reduces to a kinematically closed loop.

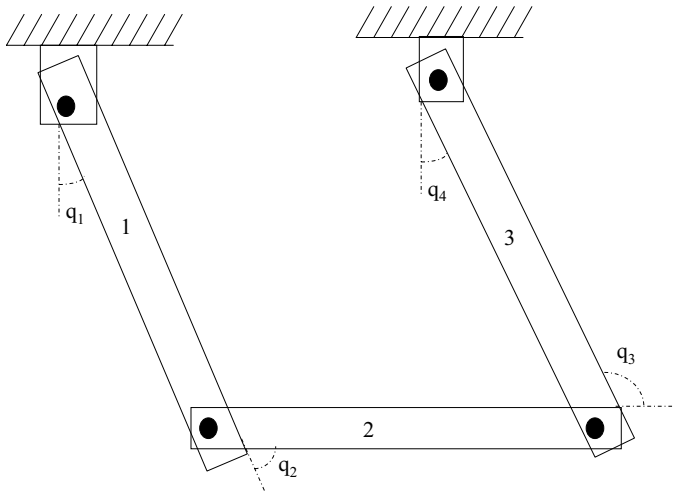


Figure 3: Simple Single Loop System

3 Procedure for Dealing with Kinematic Loops

In this section, a procedure is presented for the efficient, accurate and robust treatment of systems with kinematic loops. The procedure will first be demonstrated for a single loop system, and then will be generalized to systems containing multiple loops.

3.1 Single Loop Case

Consider a N body system connected together by kinematic joints. The base body and the terminal body of the system are connected to the inertial frame, thus forming a loop system. Figure (3) shows the general topology of this system. Other than the kinematic joints linking the base and the terminal body to the inertial frame, there is no difference between this system and a chain system. Hence, proceeding in a manner as explained in section 2, the *n-body* system can be modelled as a single assembly. The resulting two handle equations of motion of the assembly are obtained as shown below.

$$A^{bb} = \zeta_{11}^{sys} F_c^{bb} + \zeta_{12}^{sys} F_c^{tb} + \zeta_{13}^{sys} \tag{31}$$

$$A^{tb} = \zeta_{21}^{sys} F_c^{bb} + \zeta_{22}^{sys} F_c^{tb} + \zeta_{23}^{sys} \tag{32}$$

Here bb and tb denote the joints at the base body and the terminal body by which the system is connected to the inertial frame. Also, sys implies the inertia coupling terms representing the whole system. Since the system is attached to an inertial frame, the accelerations at the two ends can be given by the following kinematic relation.

$$A^{bb} = \boldsymbol{H}^{bb}\dot{u}^{bb} + \dot{\boldsymbol{H}}^{bb}u^{bb} \tag{33}$$

$$A^{tb} = \boldsymbol{H}^{tb}\dot{u}^{tb} + \dot{\boldsymbol{H}}^{tb}u^{tb} \tag{34}$$

where $\boldsymbol{H}^{bb}$ and $\boldsymbol{H}^{tb}$ represent the joint motion subspace of the joints *bb* and *tb* by which the system is connected to the inertial frame. Similarly, $\dot{u}^{bb}$ and $\dot{u}^{tb}$ represent the generalized relative accelerations at the joint degrees of freedom at *bb* and *tb*. The terms $\dot{\boldsymbol{H}}^{bb}u^{bb}$ and $\dot{\boldsymbol{H}}^{tb}u^{tb}$ represent the state dependent acceleration terms which can be kinematically calculated before solving the equations of motion.

Substituting the equations (33-34) into equations (31-32) and absorbing the state dependent acceleration

terms into the bias term, one obtains,

$$\boldsymbol{H}^{bb}\dot{u}^{bb} = \boldsymbol{\zeta}_{11}^{sys}F_c^{bb} + \boldsymbol{\zeta}_{12}^{sys}F_c^{tb} + \boldsymbol{\zeta}_{13}^{sys} \quad (35)$$

$$\boldsymbol{H}^{tb}\dot{u}^{tb} = \boldsymbol{\zeta}_{21}^{sys}F_c^{bb} + \boldsymbol{\zeta}_{22}^{sys}F_c^{tb} + \boldsymbol{\zeta}_{23}^{sys} \quad (36)$$

The two above equations (35-36) contain four unknowns F_c^{bb}, F_c^{tb}, $\dot{u}^{bb}$ and $\dot{u}^{tb}$. To eliminate two unknowns, $\dot{u}^{bb}$ and $\dot{u}^{tb}$, the kinematic relationship of the joint motion subspace and its orthogonal complement as explained by equation (9) is exploited. Multiplying the above equations by $(\boldsymbol{D}^{bb})^T$ and $(\boldsymbol{D}^{tb})^T$ respectively, where $\boldsymbol{D}^{bb}$ and $\boldsymbol{D}^{tb}$ represent the orthogonal complement of the respective joint motion subspace, one obtains

$$\overbrace{(\boldsymbol{D}^{bb})^T\boldsymbol{H}^{bb}}^{0}\dot{u}^{bb} = (\boldsymbol{D}^{bb})^T[\boldsymbol{\zeta}_{11}^{sys}F_c^{bb} + \boldsymbol{\zeta}_{12}^{sys}F_c^{tb} + \boldsymbol{\zeta}_{13}^{sys}] = 0 \quad (37)$$

$$\underbrace{(\boldsymbol{D}^{tb})^T\boldsymbol{H}^{tb}}_{0}\dot{u}^{tb} = (\boldsymbol{D}^{tb})^T[\boldsymbol{\zeta}_{21}^{sys}F_c^{bb} + \boldsymbol{\zeta}_{22}^{sys}F_c^{tb} + \boldsymbol{\zeta}_{23}^{sys}] = 0 \quad (38)$$

The above are two equations in two unknowns, viz. F_c^{bb} and F_c^{tb}. But the matrices $(\boldsymbol{D}^{bb})^T{\boldsymbol{\zeta}_{ij}}^{sys}$ and $(\boldsymbol{D}^{tb})^T{\boldsymbol{\zeta}_{ij}}^{sys}$ are rank deficient and hence these equations in their present form cannot be solved. In order to solve these equations, the constraint forces are expressed in terms of the measure numbers $\tilde{F}^{bb}$ and $\tilde{F}^{tb}$ associated with these forces as in equation (13) above.

$$F_c^{bb} = \boldsymbol{D}^{bb}\tilde{F}^{bb} \quad \text{and} \qquad F_c^{tb} = \boldsymbol{D}^{tb}\tilde{F}^{tb} \quad (39)$$

Substituting these expressions for the constraint forces into the equations (37-38) one obtains

$$(\boldsymbol{D}^{bb})^T\boldsymbol{\zeta}_{11}^{sys}\boldsymbol{D}^{bb}\tilde{F}_c^{bb} + (\boldsymbol{D}^{bb})^T\boldsymbol{\zeta}_{12}^{sys}\boldsymbol{D}^{tb}\tilde{F}_c^{tb} + (\boldsymbol{D}^{tb})^T\boldsymbol{\zeta}_{13}^{sys} = 0 \quad (40)$$

$$(\boldsymbol{D}^{tb})^T\boldsymbol{\zeta}_{21}^{sys}\boldsymbol{D}^{bb}\tilde{F}_c^{bb} + (\boldsymbol{D}^{tb})^T\boldsymbol{\zeta}_{22}^{sys}\boldsymbol{D}^{tb}\tilde{F}_c^{tb} + (\boldsymbol{D}^{tb})^T\boldsymbol{\zeta}_{23}^{sys} = 0 \quad (41)$$

In these equations, the terms $(\boldsymbol{D}^{bb})^T{\boldsymbol{\zeta}_{11}}^{sys}\boldsymbol{D}^{bb}$ and $(\boldsymbol{D}^{tb})^T{\boldsymbol{\zeta}_{22}}^{sys}\boldsymbol{D}^{tb}$ are symmetric positive definite (SPD) matrices and there is no problem associated with their inversion. For notational convenience, the above equations can be represented compactly in matrix form as

$$\begin{bmatrix}\boldsymbol{\chi}_{11} & \boldsymbol{\chi}_{12}\\ \boldsymbol{\chi}_{21} & \boldsymbol{\chi}_{22}\end{bmatrix}\begin{bmatrix}\tilde{F}_c^{bb}\\ \tilde{F}_c^{tb}\end{bmatrix} = -\begin{bmatrix}\boldsymbol{\chi}_{13}\\ \boldsymbol{\chi}_{23}\end{bmatrix} \quad (42)$$

where the corresponding χ_{ij} can be derived from above equation. The matrix in (42) is also SPD with $\chi_{12} = \chi_{21}^T$. Having solved the above equations for the values of $\tilde{F}_c^{bb}$ and $\tilde{F}_c^{tb}$, the corresponding expression for F_c^{bb} and F_c^{tb} can be obtained by pre-multiplying the corresponding expressions by $(\boldsymbol{D}^{bb})$ and $(\boldsymbol{D}^{tb})$ respectively as shown in equation (39).

At this point, both constraint forces on terminal and base joints are known. Consequently, the two handle equations of motion of the single assembly can be solved to obtain the spatial accelerations at the corresponding joints. This initiates the hierarchic disassembly process discussed above which successively calculates the spatial accelerations and constraint forces of sub-assemblies. This disassembly results in the spatial accelerations and constraint forces calculated on each physical body in the system.

3.2 Multiple Closed Loops

The treatment of coupled closed loops is presented in this section. The methodology presented here is applicable for cases where there are multiple loops as well as for cases when a single loop is connected to a chain structure. Consider the double loop system shown in figure (4). As seen in the figure, bodies $1, 2, 3$ form the upper loop while bodies $4, 5, 6$ form the lower loop with body 2 being the common body shared between the two loops. The objective is to reduce the lower loop into a single assembly by coupling together the two handle equations of motion of bodies $2, 4, 5$ and 6. This single assembly and the bodies in the upper loop then form a single loop system

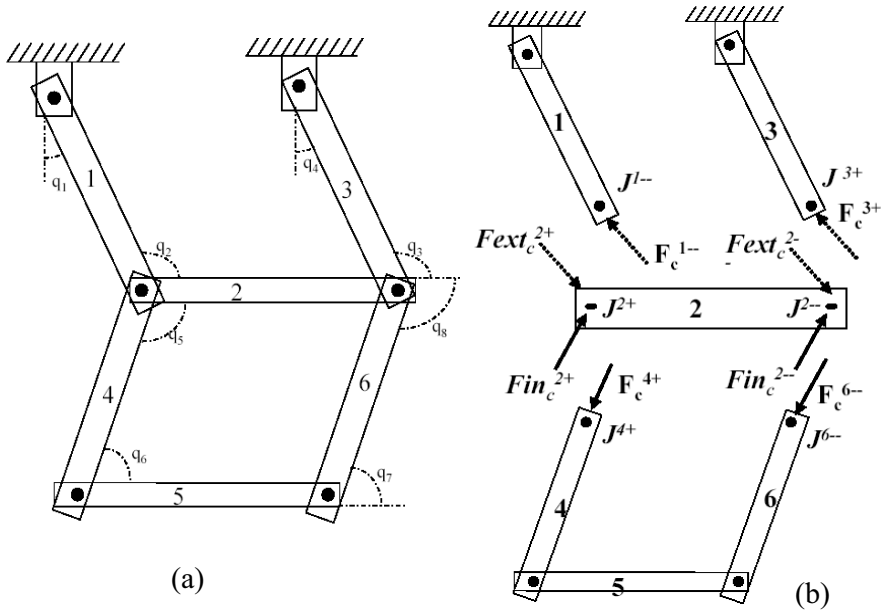


Figure 4: Coupled Loop System

Proceeding in a manner similar to a chain system, the bodies $4, 5$ and 6 can be coupled together to form the two handle equations of motion of the resulting system as below. In this case the joints 4^+ and 6^- are the base and terminal joints for the system *sys* representing the assembly of bodies 4,5, and 6 while ζ_{ij}^{sys} represent the inertia coupling terms of the assembly.

$$A^{4^+} = \zeta_{11}^{sys} F_c^{4^+} + \zeta_{12}^{sys} F_c^{6^-} + \zeta_{13}^{sys} \tag{43}$$

$$A^{6^-} = \zeta_{21}^{sys} F_c^{4^+} + \zeta_{22}^{sys} F_c^{6^-} + \zeta_{23}^{sys} \tag{44}$$

Similarly, the two handle equations of motion of the body 2 can be written as below.

$$A^{2^+} = \zeta_{11}^{2} F_c^{2+} + \zeta_{12}^{2} F_c^{2^-} + \zeta_{13}^{2} \tag{45}$$

$$A^{2^-} = \zeta_{21}^{2} F_c^{2+} + \zeta_{22}^{2} F_c^{2^-} + \zeta_{23}^{2} \tag{46}$$

where 2^+ is the joint between bodies 2 and 1 while 2^- is the joint between bodies 2 and 3.

For body 2, the constraint forces F_c^{2+} and F_c^{2-} can be further analyzed as

$$F_c^{2+} = Fin_c^{2+} + Fext_c^{2+} \tag{47}$$

$$F_c^{2-} = Fin_c^{2-} + Fext_c^{2-} \tag{48}$$

Here Fin represents the force acting on body 2 which originates from the lower loop. This force is an internal force when the lower loop is considered as a single system. $Fext$ is the force acting on body 2 due to the interactions with bodies 1 and 3. Thus if the lower loop were to be represented as a single assembly with handles at joints $2+$ and $2-$, the terms represented by Fin would disappear and the terms $Fext$ would represent the constraint forces at the two handles of the resulting assembly. The objective thus is to couple

the equations of the resulting assembly of bodies $4, 5, 6$ with the equations of body 2 to get the two handle equations of the resulting assembly. This assembly be referred to as asm henceforth.

Consider the kinematic expression for the accelerations of the two handles of the assembly of bodies $4, 5, 6$. These can be written as

$$A^{4^+} = A^{2+} + \boldsymbol{H}^{4^+}\dot{u}^{4^+} + \dot{\boldsymbol{H}}^{4^+}u^{4^+} \tag{49}$$

$$A^{6^-} = A^{2-} + \boldsymbol{H}^{6^-}\dot{u}^{6^-} + \dot{\boldsymbol{H}}^{6^-}u^{6^-} \tag{50}$$

Also note that from Newton's second law,

$$F_c^{4^+} = -Fin_c^{2+} \tag{51}$$

$$F_c^{6^-} = -Fin_c^{2-} \tag{52}$$

Substituting the expressions of A^{4^+} from equation (49) and A^{6^-} from equation (50) as well as the expressions of constraint forces from equation (51-52) into the corresponding two handle equations of motion (43-44), we obtain

$$A^{4^+} = A^{2+} + \boldsymbol{H}^{4^+}\dot{u}^{4^+} + \dot{\boldsymbol{H}}^{4^+}u^{4^+} = -\boldsymbol{\zeta}_{11}^{sys}Fin_c^{2+} - \boldsymbol{\zeta}_{12}^{sys}Fin_c^{2-} + \boldsymbol{\zeta}_{13}^{sys} \tag{53}$$

$$A^{6^-} = A^{2-} + \boldsymbol{H}^{6^-}\dot{u}^{6^-} + \dot{\boldsymbol{H}}^{6^-}u^{6^-} = -\boldsymbol{\zeta}_{21}^{sys}Fin_c^{2+} - \boldsymbol{\zeta}_{22}^{sys}Fin_c^{2-} + \boldsymbol{\zeta}_{23}^{sys} \tag{54}$$

Subtracting equation (53) from equation (45) and similarly equation (54) from equation (46), an expression for the relative joint acceleration can be obtained as

$$\begin{aligned} -\boldsymbol{H}^{4^+}\dot{u}^{4^+} &= \boldsymbol{\zeta}_{11}^2 Fext_c^{2+} + [\boldsymbol{\zeta}_{11}^2 + \boldsymbol{\zeta}_{11}^{sys}]Fin_c^{2+} + \\ &\quad \boldsymbol{\zeta}_{12}^2 Fext_c^{2-} + [\boldsymbol{\zeta}_{12}^2 + \boldsymbol{\zeta}_{12}^{sys}]Fin_c^{2-} + [\boldsymbol{\zeta}_{13}^2 - \boldsymbol{\zeta}_{13}^{sys} + \dot{\boldsymbol{H}}^{4^+}u^{4^+}] \end{aligned} \tag{55}$$

$$\begin{aligned} -\boldsymbol{H}^{6^-}\dot{u}^{6^-} &= \boldsymbol{\zeta}_{21}^2 Fext_c^{2+} + [\boldsymbol{\zeta}_{21}^2 + \boldsymbol{\zeta}_{21}^{sys}]Fin_c^{2+} + \\ &\quad \boldsymbol{\zeta}_{22}^2 Fext_c^{2-} + [\boldsymbol{\zeta}_{22}^2 + \boldsymbol{\zeta}_{22}^{sys}]Fin_c^{2-} + [\boldsymbol{\zeta}_{23}^2 - \boldsymbol{\zeta}_{23}^{sys} + \dot{\boldsymbol{H}}^{6^-}u^{6^-}] \end{aligned} \tag{56}$$

The orthogonal complement $\boldsymbol{D}^{4^+}$ and $\boldsymbol{D}^{6^-}$ are orthogonal to the joint motion subspace matrices $\boldsymbol{H}^{4^+}$ and $\boldsymbol{H}^{6^-}$. Hence premultiplying above equations (55) and (56) by $(\boldsymbol{D}^{4+})^T$ and $(\boldsymbol{D}^{6-})^T$ and using the relation from equation (9), the following can be arrived at.

$$\begin{aligned} -(\boldsymbol{D}^{4^+})^T\boldsymbol{H}^{4^+}\dot{u}^{4^+} &= 0 = (\boldsymbol{D}^{4^+})^T\boldsymbol{\zeta}_{11}^2 Fext_c^{2+} - (\boldsymbol{D}^{4^+})^T[\boldsymbol{\zeta}_{11}^2 + \boldsymbol{\zeta}_{11}^{sys}]F_c^{4^+} + (\boldsymbol{D}^{4^+})^T\boldsymbol{\zeta}_{12}^2 Fext_c^{2-} \\ &\quad -(\boldsymbol{D}^{4^+})^T[\boldsymbol{\zeta}_{12}^2 + \boldsymbol{\zeta}_{12}^{sys}]F_c^{6^-} + (\boldsymbol{D}^{4^+})^T[\boldsymbol{\zeta}_{13}^2 - \boldsymbol{\zeta}_{13}^{sys} + \dot{\boldsymbol{H}}^{4^+}u^{4^+}] \end{aligned} \tag{57}$$

$$\begin{aligned} -(\boldsymbol{D}^{6^-})^T\boldsymbol{H}^{6^-}\dot{u}^{6^-} &= 0 = (\boldsymbol{D}^{6^-})^T\boldsymbol{\zeta}_{21}^2 Fext_c^{2+} - (\boldsymbol{D}^{6^-})^T[\boldsymbol{\zeta}_{21}^2 + \boldsymbol{\zeta}_{21}^{sys}]F_c^{4^+} + (\boldsymbol{D}^{6^-})^T\boldsymbol{\zeta}_{22}^2 Fext_c^{2-} \\ &\quad -(\boldsymbol{D}^{6^-})^T[\boldsymbol{\zeta}_{22}^2 + \boldsymbol{\zeta}_{22}^{sys}]F_c^{6^-} + (\boldsymbol{D}^{6^-})^T[\boldsymbol{\zeta}_{23}^2 - \boldsymbol{\zeta}_{23}^{sys} + \dot{\boldsymbol{H}}^{6^-}u^{6^-}] \end{aligned} \tag{58}$$

Further note that the constraint forces can be expressed in terms of the measure numbers and the orthogonal complement of the joint motion subspace i.e.

$$F_c^{4^+} = \boldsymbol{D}^{4^+}\tilde{F}^{4^+} \qquad \text{and} \qquad F_c^{6^-} = \boldsymbol{D}^{6^-}\tilde{F}^{6^-} \tag{59}$$

From these, equations (57) and (58) can be manipulated to generate an expression for the internal forces viz. $Fin_c^{2^+}$ and $Fin_c^{2^-}$ as shown below.

$$(\boldsymbol{D}^{4^+})^T[\boldsymbol{\zeta}_{11}^2+\boldsymbol{\zeta}_{11}^{sys}]F_c^{4^+}+(\boldsymbol{D}^{4^+})^T[\boldsymbol{\zeta}_{12}^2+\boldsymbol{\zeta}_{12}^{sys}]F_c^{6^-}=$$
$$(\boldsymbol{D}^{4^+})^T\boldsymbol{\zeta}_{11}^2 Fext_c^{2+}+(\boldsymbol{D}^{4^+})^T\boldsymbol{\zeta}_{12}^2 Fext_c^{2-}+(\boldsymbol{D}^{4^+})^T[\boldsymbol{\zeta}_{13}^2-\boldsymbol{\zeta}_{13}^{sys}+\dot{\boldsymbol{H}}^{4^+}u^{4^+}] \quad (60)$$
$$\Rightarrow (\boldsymbol{D}^{4^+})^T[\boldsymbol{\zeta}_{11}^2+\boldsymbol{\zeta}_{11}^{sys}]\boldsymbol{D}^{4^+}\tilde{F}^{4^+}+(\boldsymbol{D}^{4^+})^T[\boldsymbol{\zeta}_{12}^2+\boldsymbol{\zeta}_{12}^{sys}]\boldsymbol{D}^{6^-}\tilde{F}^{6^-}=$$
$$(\boldsymbol{D}^{4^+})^T\boldsymbol{\zeta}_{11}^2 Fext_c^{2+}+(\boldsymbol{D}^{4^+})^T\boldsymbol{\zeta}_{12}^2 Fext_c^{2-}+(\boldsymbol{D}^{4^+})^T[\boldsymbol{\zeta}_{13}^2-\boldsymbol{\zeta}_{13}^{sys}+\dot{\boldsymbol{H}}^{4^+}u^{4^+}] \quad (61)$$

$$(\boldsymbol{D}^{6^-})^T[\boldsymbol{\zeta}_{21}^2+\boldsymbol{\zeta}_{21}^{sys}]F_c^{4^+}+(\boldsymbol{D}^{6^-})^T[\boldsymbol{\zeta}_{22}^2+\boldsymbol{\zeta}_{22}^{sys}]F_c^{6^-}=$$
$$(\boldsymbol{D}^{6^-})^T\boldsymbol{\zeta}_{21}^2 Fext_c^{2+}+(\boldsymbol{D}^{6^-})^T\boldsymbol{\zeta}_{22}^2 Fext_c^{2-}+(\boldsymbol{D}^{6^-})^T[\boldsymbol{\zeta}_{23}^2-\boldsymbol{\zeta}_{23}^{sys}+\dot{\boldsymbol{H}}^{6^-}u^{6^-}] \quad (62)$$
$$\Rightarrow (\boldsymbol{D}^{6^-})^T[\boldsymbol{\zeta}_{21}^2+\boldsymbol{\zeta}_{21}^{sys}]\boldsymbol{D}^{4^+}\tilde{F}^{4^+}+(\boldsymbol{D}^{6^-})^T[\boldsymbol{\zeta}_{22}^2+\boldsymbol{\zeta}_{22}^{sys}]\boldsymbol{D}^{6^-}\tilde{F}^{6^-}=$$
$$(\boldsymbol{D}^{6^-})^T\boldsymbol{\zeta}_{21}^2 Fext_c^{2+}+(\boldsymbol{D}^{6^-})^T\boldsymbol{\zeta}_{22}^2 Fext_c^{2-}+(\boldsymbol{D}^{6^-})^T[\boldsymbol{\zeta}_{23}^2-\boldsymbol{\zeta}_{23}^{sys}+\dot{\boldsymbol{H}}^{6^-}u^{6^-}] \quad (63)$$

In matrix format, the above equations (60-63) can be expressed as

$$\begin{bmatrix}\tilde{F}^{4^+}\\ \tilde{F}^{6^-}\end{bmatrix} = [\boldsymbol{X}]\,[\boldsymbol{Y}]\begin{bmatrix}Fext_c^{2+}\\ Fext_c^{2-}\end{bmatrix}+[\boldsymbol{X}]\,[\boldsymbol{K}] \quad (64)$$

$$\Rightarrow \begin{bmatrix}F_c^{4^+}\\ F_c^{6^-}\end{bmatrix} = -\begin{bmatrix}Fin_c^{2^+}\\ Fin_c^{2^-}\end{bmatrix} = [\boldsymbol{D}]\,[\boldsymbol{X}]\,[\boldsymbol{Y}]\begin{bmatrix}Fext_c^{2+}\\ Fext_c^{2-}\end{bmatrix}+[\boldsymbol{D}]\,[\boldsymbol{X}]\,[\boldsymbol{K}] \quad (65)$$

$$\text{where } [\boldsymbol{X}] = \begin{bmatrix}(\boldsymbol{D}^{4^+})^T[\boldsymbol{\zeta}_{11}^2+\boldsymbol{\zeta}_{11}^{sys}]\boldsymbol{D}^{4^+} & (\boldsymbol{D}^{4^+})^T[\boldsymbol{\zeta}_{12}^2+\boldsymbol{\zeta}_{12}^{sys}]\boldsymbol{D}^{6^-}\\ (\boldsymbol{D}^{6^-})^T[\boldsymbol{\zeta}_{21}^2+\boldsymbol{\zeta}_{21}^{sys}]\boldsymbol{D}^{4^+} & (\boldsymbol{D}^{6^-})^T[\boldsymbol{\zeta}_{22}^2+\boldsymbol{\zeta}_{22}^{sys}]\boldsymbol{D}^{6^-}\end{bmatrix}^{-1} \quad (66)$$

$$\text{and } [\boldsymbol{Y}] = \begin{bmatrix}(\boldsymbol{D}^{4^+})^T\boldsymbol{\zeta}_{11}^2 & (\boldsymbol{D}^{4^+})^T\boldsymbol{\zeta}_{12}^2\\ (\boldsymbol{D}^{6^-})^T\boldsymbol{\zeta}_{21}^2 & (\boldsymbol{D}^{6^-})^T\boldsymbol{\zeta}_{22}^2\end{bmatrix} \quad (67)$$

$$\text{and } [\boldsymbol{K}] = \begin{bmatrix}(\boldsymbol{D}^{4^+})^T[\boldsymbol{\zeta}_{13}^2-\boldsymbol{\zeta}_{13}^{sys}+\dot{\boldsymbol{H}}^{4^+}u^{4^+}]\\ (\boldsymbol{D}^{6^-})^T[\boldsymbol{\zeta}_{23}^2-\boldsymbol{\zeta}_{23}^{sys}+\dot{\boldsymbol{H}}^{6^-}u^{6^-}]\end{bmatrix} \quad (68)$$

$$\text{and } [\boldsymbol{D}] = \begin{bmatrix}\boldsymbol{D}^{4^+} & \boldsymbol{Z}\\ \boldsymbol{Z} & \boldsymbol{D}^{6^-}\end{bmatrix} \quad (69)$$

Substituting the expression for $Fin_c^{2^+}$ and $Fin_c^{2^-}$ in the two handle equations of motion for body 2, the two handle equations of motion of the entire assembly of bodies 4,5,6 and 2 can be obtained as

$$\begin{bmatrix}A^{2^+}\\ A^{2^-}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\zeta}_{11}^{asm} & \boldsymbol{\zeta}_{12}^{asm}\\ \boldsymbol{\zeta}_{21}^{asm} & \boldsymbol{\zeta}_{22}^{asm}\end{bmatrix}\begin{bmatrix}Fext_c^{2+}\\ Fext_c^{2-}\end{bmatrix}+\begin{bmatrix}\boldsymbol{\zeta}_{13}^{asm}\\ \boldsymbol{\zeta}_{23}^{asm}\end{bmatrix} \quad (70)$$

$$\text{where } \begin{bmatrix}\boldsymbol{\zeta}_{11}^{asm} & \boldsymbol{\zeta}_{12}^{asm}\\ \boldsymbol{\zeta}_{21}^{asm} & \boldsymbol{\zeta}_{22}^{asm}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\zeta}_{11}^2 & \boldsymbol{\zeta}_{12}^2\\ \boldsymbol{\zeta}_{21}^2 & \boldsymbol{\zeta}_{22}^2\end{bmatrix}([\boldsymbol{U}]-[\boldsymbol{D}]\,[\boldsymbol{X}]\,[\boldsymbol{Y}]) \quad (71)$$

$$\text{and } \begin{bmatrix}\boldsymbol{\zeta}_{13}^{asm}\\ \boldsymbol{\zeta}_{23}^{asm}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\zeta}_{13}^2\\ \boldsymbol{\zeta}_{23}^2\end{bmatrix}-[\boldsymbol{D}]\,[\boldsymbol{X}]\,[\boldsymbol{K}] \quad (72)$$

This body asm now represents a single body in the upper loop. The upper loop now is made up of bodies 1, asm and 3. This upper loop now reduces to a single loop and can be solved exactly as in the above section. Having solved for the accelerations at the handles of each body in the upper loop, the same procedure can be applied for the lower loop to generate the acceleration of each handle on each body of the lower loop.

3.3 Singular Configurations

Multibody systems with closed kinematical loops can undergo motion such that the system passes through a singular configuration. Singular configurations are typically observed when system enters some form of a toggle position. Example of such cases can be a four bar linkage as shown in figure (5).

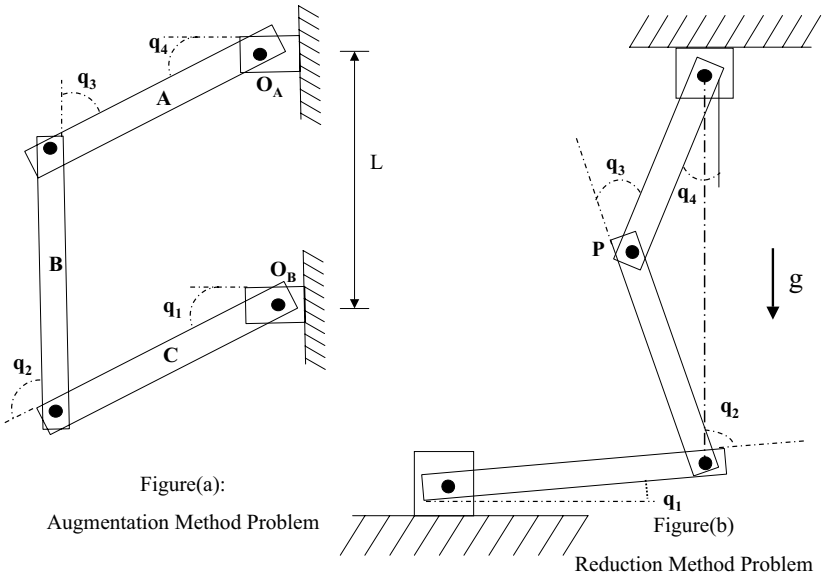


Figure 5: Four Bar Linkages for simulating Singular Configurations

When using an augmentation approach for formulating the equations of motion, a common problem encountered with the system in a singular configuration is that the constraint Jacobian becomes rank deficient. A rank deficient constraint Jacobian reduces the system of equations to unsolvable, rendering the formulation incapable of handling such configurations. A similar problem is seen with a reduction type of approach where the *dependency* matrix loses rank when the system enters a singular configuration. In either case, in a true singular configuration, the system of equations cannot be solved. Even if the system does not enter a true singular configuration, but passes near singular states, constraint violation errors can grow significantly due to the ill conditioned constraint Jacobian or dependency matrix. This is often encountered as the integration steps across a singular configuration during a simulation and can result in significant errors in the simulation.

The algorithm presented in this paper is able to simulate systems with singular configurations without running into these problem. This is because neither does the formulation construct a constraint matrix nor does it use dependent and independent coordinates. Thus, since the dimensionality of the problem never changes, the algorithm is free from rank deficiency issues with all matrices to be inverted remaining positive definite.

The ODCA algorithm presented uses a redundant set of generalized coordinates, but does not carry along a companion set of algebraic constraint equations. There is thus no constraint Jacobian to lose rank, but individual joint constraints are enforced implicitly through the joint space map H (**??**q (8)). This manner in which this method avoids singularities appears similar in some regards to that with Euler parameters. With Euler parameter, one deals with a redundant four member set of generalized coordinates (parameters) for the global and nonsingular description of three degree of freedom spatial rotation. The constraint between these four coordinates being implicitly enforced. If the constraint were explicitly used to reduce out the extra generalized coordinate (parameter), the representation again may become singular.

4 Numerical Examples

In this section numerical results obtained from implementing sample test cases are presented. These test cases were run with the intent of assessing the basic characteristics of the presented ODCA approach, relative to those obtained using more traditional methods. For this reason simulations were run using: 1) the

presented ODCA method; 2) constraint enforcement at the acceleration level, via Lagrange multipliers; and 3) Constraint enforcement at the velocity level by removing the redundant variables at both the acceleration and velocity levels [34]. Thus, the following sample case were all run without the benefit of any form of supplemental constraint stabilization. In this manner the relative merit of these methods with regard to accuracy and robustness might be more clearly seen. All the test cases presented here were implemented in Matlab™ and the temporal integration of the equations of motion were carried out using the $ode45$ integrator featured in Matlab™. The absolute and relative tolerances were set to 10^{-8}. All bodies in the test cases are modelled as having length L = 1, with mass of $1kg$ and inertia about an axis perpendicular to the page of the paper as $1kgm^2$. All kinematic joints in the test cases are revolute.

4.1 Single and Coupled Loops

Consider the four bar linkage as shown in figure (3). The following figure (6) shows the results obtained from a simulation of this system when moving under the effect of gravity. Figure (6) shows the variation in the angles with respect to time while figure (7) shows the variation of the constraint violation error with time. The constraint violation error refers to the absolute position error in the satisfaction of the loop closer equation. As can be seen in the error plot, the error is of the order of 10^{-14} i.e. up to machine accuracy. A similar test case is shown for systems with coupled loops. Consider the seven bar linkage as shown in figure(4). The results of a 10 second simulation of the system released from rest under the effect of gravity are shown in figure (9). The constraint violation is plotted in figure(8). The constraint violation in this case is also of the order of 10^{-14} i.e. up to machine accuracy.

Analysis of this error time history indicates that after the first few seconds of the simulation, the constraint violation error grows as $t^{0.7}$, without any additional constraint stabilization strategy being used. This represents a considerable improvement in performance over conventional constraint enforcement methods. Specifically, when this same system is modelled through the enforcement of the loop closure constraints at the acceleration level alone via Lagrange multipliers, the the error increases as t^2. Similarly, if the constraints for this system are enforced a the velocity level, through the elimination of redundant variables, then the constraint violation error grows linearly with t. In the single loop, and multiple loop configurations investigated, the presented ODCA procedure consistently showed such reduced constraint violation error growth relative to more traditional methods.

4.2 Singular Configurations

Two test cases are simulated to compare the performance of this algorithm for systems repeatedly passing through singular configurations. These test cases are shown in Figure (5). For each of these systems 20 second simulations were carried out, with the systems being acted on by gravity, and released from rest from a non equilibrium position. In each case, the system equilibrium position, about which the systems oscillate produces a numerical singularity. As such these test cases offer a challenge to traditional constraint enforcement methods. When the function evaluations occur near the singular configuration, a significant loss of accuracy is incurred due to the ill conditioning of the system. If the function evaluation occurs on (or too near) a singular configuration, then these traditional methods will fail completely. The results are shown in Figure (10).

For the system shown in Figure (5-a), the simulation results obtained with the present algorithm are compared with a reference simulation generated using an augmented Lagrange approach. When using the augmented Lagrange approach, the constraint Jacobian for this system loses rank when the coordinate q_1 becomes zero i.e. $q_1 = 0$. In this configuration there are three possible solutions i.e. *a)* the system passes though the vertical and remains in a parallelogram configuration *b)* the systems jumps to the 'anti-parallelogram' (crossed) circuit of motion or *c)* bar A becomes kinematically locked in the vertical direction as bars B and C swing as a unit as a pendulum about hinge O_B.

Similarly, for the system shown in Figure(5-b), the simulation using the O-DCA is compared against a reference simulation generated using a reduction approach. In the reduction approach $\ddot{q_1}$ is selected as the independent acceleration variable with $\ddot{q_2}$, $\ddot{q_3}$, $\ddot{q_4}$ as the dependent acceleration variables. With this system, the dependency matrix prescribing $\ddot{q_2}$, $\ddot{q_3}$, $\ddot{q_4}$ as a function of $\ddot{q_1}$ becomes singular as the system enters the

toggle position at $q_4 = 90^\circ$. In this configuration the dependency equations offer two solutions with a further increase in q_1 i.e. *a)* hinge point P will move to the left or *b)* move to the right.

In all case the presented ODCA method significantly out performed the traditional Lagrange multiplier and reduction approaches. The constraint violation error time histories for each of these are presented in figures 10. The relative robustness and accuracy of the presented ODCA was particularly evident when for each of these mechanisms, the systems was started with an nonzero initial velocity in a truly singular configuration. The ODCA was unaffected and performed the simulation without difficulty, while the more traditional constraint enforcement methods all failed immediately and completely.

4.3 Discussion

When simulating dynamics of systems with constraints, the constraint are most often imposed at the accelerations, or possibly the velocity level. The constraint imposition at acceleration or velocity level introduces zero eigenvalues associated with each constant of temporal integration. This problem is further exacerbated with accumulation of round-off errors from finite precision arithmetic. Thus, imposing the constraints at the acceleration or velocity level results in an eventual unstable growth in constraint violation which can grow exponentially for a given simulation. This problem can be alleviated by introducing some form of constraint stabilization into the equations. Although introducing a constraint stabilization technique can reduce the drift in the constraint violation significantly, these methods do not generally provide full constraint satisfaction, and come with their own (potentially significant) computational cost. The onus is thus on the underlying formulation to keep the constraint violation at a minimum.

The results obtained from the implementation of the algorithm for sample test cases are indicative of the excellent performance of this algorithm for systems with single or coupled loops. Although the constraints are imposed at the acceleration level, the constraint violation error growth achieved with the ODCA was consistently superior to that obtained using even velocity level constraint enforcement with more traditional constraint enforcement methods. Even though there is a small growth in the error, the magnitude of the error is far smaller than what may be expected for a comparable length simulation using acceleration level constraint imposition with either augmented or reduction approach. The error can be further reduced with the use of a constraint stabilization technique or by imposing the constraint at the velocity level.

For systems which pass through singular configuration, the performance of the present algorithm is far superior to that of a true augmentation approach as well as a reduction type approach with the constraint imposition at the acceleration level. For the test cases simulated, the acceleration level augmented approach failed to converge, while the reduction approach showed a large error in constraint violation. The performance of the present algorithm is comparable with the reduction approach with a velocity level constraint imposition. In fact, for the system simulated, the present algorithm gives slightly better results. Simulations run with both the reduction and augmentation approaches failed when the system entered a true singular configuration. The results shown here for these approaches are for simulations when the system does not enter a true singular configuration but passes through it. By comparison, the method presented here continued to run correctly even when it entered a truly singular configuration (not just near singular).

5 Conclusion

In this paper, a new algorithm is presented for calculating the forward dynamics of multibody systems with closed kinematical loops. The algorithm is simple to implement and the computational expense is $O(n)$ when implemented in serial. Due to the binary-tree structure of the formulation, the algorithm would achieve a $logarithmic$ complexity for parallel implementation. The algorithm is exact and non-iterative. The implementation results indicate excellent constraint satisfaction for systems with single as well as coupled loops, even when the constraint satisfaction is enforced at the acceleration level. The performance of the algorithm is better than comparable algorithms for modelling systems which pass through singular configurations. The algorithm can be extended for flexible body systems modelled using a component mode synthesis formulation. The implementation of the constraints at the velocity level using this algorithm as well as performance measures for larger systems are of current research interest for the authors.

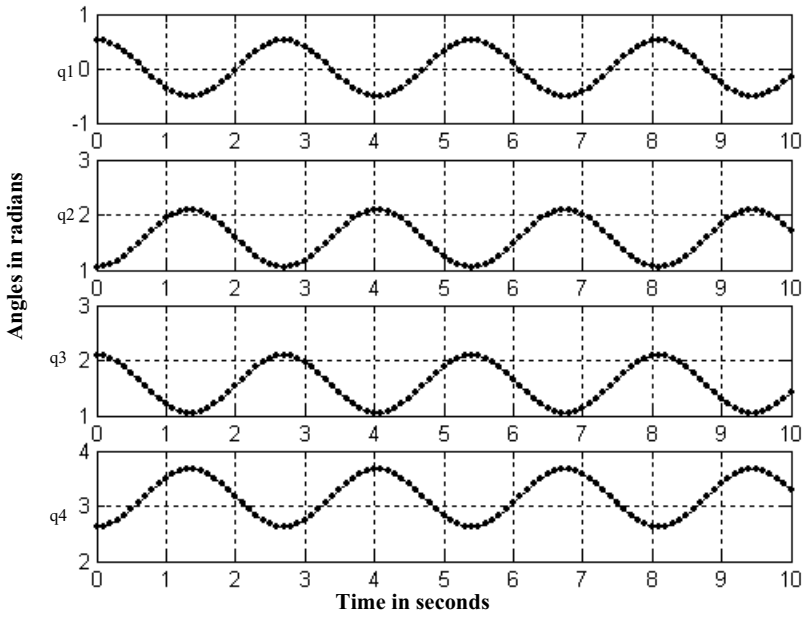


Figure 6: Results for Single Loop

6 Acknowledgement

This work was funded by the NSF NIRT Grant Number 0303902. The authors would like to thank the funding agency for their support.

References

[1] W. W. Hooker and G. Margulies. The dynamical attitude equations for an n-body satellite. *Journal of the Astronautical Sciences*, 7(4):123–128, Winter 1965.

[2] J. S. Y. Luh, M. W. Walker, and R. P. C. Paul. On-line computational scheme for mechanical manipulators. *Journal of Dynamic Systems, Measurements, and Control*, 102:69–76, Jun. 1980.

[3] M. W. Walker and D. E. Orin. Efficient dynamic computer simulation of robotic mechanisms. *Journal of Dynamic Systems, Measurement, and Control*, 104:205–211, Sep. 1982.

[4] D. E. Rosenthal and M. A. Sherman. High performance multibody simulations via symbolic equation manipulation and Kane's method. *Journal of the Astronautical Sciences*, 34(3):223–239, Jul.–Sep. 1986.

[5] P. E. Neilan. *Efficient Computer Simulation of Motions of Multibody Systems*. PhD thesis, Stanford University, 1986.

[6] A. F. Vereshchagin. Computer simulation of the dynamics of complicated mechanisms of robot-manipulators. *Engineering Cybernetics*, 12(6):65–70, November–December 1974.

[7] W. W. Armstrong. Recursive solution to the equations of motion of an n-link manipulator. In *Fifth World Congress on the Theory of Machines and Mechanisms*, volume 2, pages 1342–1346, 1979.

[8] R. Featherstone. The calculation of robotic dynamics using articulated body inertias. *International Journal of Robotics Research*, 2(1):13–30, Spring 1983.

[9] 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 *IFAC/IFIP/IMACS Symposium*, Vienna, Austria, 1986.

[10] D. S. Bae and E. J. Haug. A recursive formation for constrained mechanical system dynamics: Part I, Open loop systems. *Mechanisms, Structures, and Machines*, 15(3):359–382, 1987.

[11] D.E. Rosenthal. An order n formulation for robotic systems. *The Journal of the Astronautical Sciences*, 38(4):511–529, 1990.

[12] K. S. Anderson. *Recursive Derivation of Explicit Equations of Motion for Efficient Dynamic/Control Simulation of Large Multibody Systems*. PhD thesis, Stanford University, 1990.

[13] K. Kreutz-Delgado, A. Jain, and G. Rodriguez. Recursive formulation of operational space control. *International Journal of Robotics Research*, 11(4):320–328, August 1992.

[14] A. Jain. Unified formulation of dynamics for serial rigid multibody systems. *Journal of Guidance, Control, and Dynamics*, 14(3):531–542, May–Jun. 1991.

[15] R. Featherstone. *Robot Dynamics Algorithms*. Kluwer Academic Publishing, New York, 1987.

[16] D. S. Bae and E. J. Haug. A recursive formation for constrained mechanical system dynamics: Part II, Closed loop systems. *Mechanisms, Structures, and Machines*, 15(4):481–506, 1987.

[17] R. A. Wehage. Application of matrix partitioning and recursive projection to O(n) solution of constrained equations of motion. American Society of Mechanical Engineers, Design Engineering Division (Publication) DE, v15-2, pages 221–230. 1988.

[18] K. S. Anderson and J. H. Critchley. Improved order-n performance algorithm for the simulation of constrained multi-rigid-body systems. *Multibody Systems Dynamics*, 9:185–212, 2003.

[19] J. H. Critchley and K. S. Anderson. A generalized recursive coordinate reduction method for multibody system dynamics. *Journal of Multiscale Computational Engineering*, 1(2 & 3):181–200, 2003.

[20] P. Lötstedt. On a penalty function method for the simulation of mechanical systems subject to constraints. Report TRITA-NA-7919, Royal Institute of Technology, Stockholm, Sweden, 1979.

[21] P. Lötstedt. Mechanical systems of rigid bodies subject to unilateral constraints. *SIAM Journal of Applied Mathematics*, 42(2):281–296, Apr. 1982.

[22] J. Baumgarte. Stabilization of constraints and integrals of motion in dynamic systems. *Computer Methods in Applied Mechanics and Engineering*, 1:1–16, 1972.

[23] J. W. Baumgarte. A new method for stabilization of holonomic contraints. *Journal of Applied Mechanics*, 50:869–870, Dec. 1983.

[24] K. C. Park and J. C. Chiou. Stabilization of computational procedures for constrained dynamical systems. *Journal of Guidance, Control and Dynamics*, 11(4):365–370, Jul.–Aug. 1988.

[25] E. Bayo, J. Garcia de Jalon, and M. A. Serna. Modified Lagrangian formulation for the dynamic analysis of constrained mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 71(2):183–195, Nov. 1988.

[26] K. S. Anderson. An order-n formulation for motion simulation of general constrained multi-rigid-body systems. *Computers and Structures*, 43(3):565–572, 1992.

[27] V. Stejskal and M. Valášek. *Kinematics and Dynamics of Machinery*. Marcel Dekker, New York, NY, 1996.

[28] S. K. Saha and W. O. Schiehlen. Recursive kinematics and dynamics for parallel structured closed-loop multibody systems. *Mechanical Structures and Machines*, 29(2):143–175, 2001.

[29] A. Fijany, I. Sharf, and G. M. T. D'Eleuterio. Parallel O(log n) algorithms for computation of manipulator forward dynamics. *IEEE Transactions on Robotics and Automation*, 11(3):389–400, 1995.

[30] R. Featherstone. A divide-and-conquer articulated body algorithm for parallel $O(\log(n))$ calculation of rigid body dynamics. Part 1: Basic algorithm. *International Journal of Robotics Research*, 18(9):867–875, Sep. 1999.

[31] R. Featherstone. A divide-and-conquer articulated body algorithm for parallel $O(\log(n))$ calculation of rigid body dynamics. Part 2: Trees, loops, and accuracy. *International Journal of Robotics Research*, 18(9):876–892, September 1999.

[32] Kim S. S., and VanderPloeg M. J., 1986. "Generalized and Efficient Method for Dynamic Analysis of Mechanical Systems Using Velocity Transforms". Journal of Mechanism, Transmissions, and Automation Design, **108** (2), pp. 176–182.

[33] Nikravesh P. E., 1990. "Systematic Reduction of Multibody Equations to a Minimal Set". International Journal of Non-Linear Mechanics, **25** (2-3), pp. 143–151.

[34] Kane T. R. and Levinson D. A., 1985. "Dynamics: Theory and Application". Mcgraw-Hill, NY.

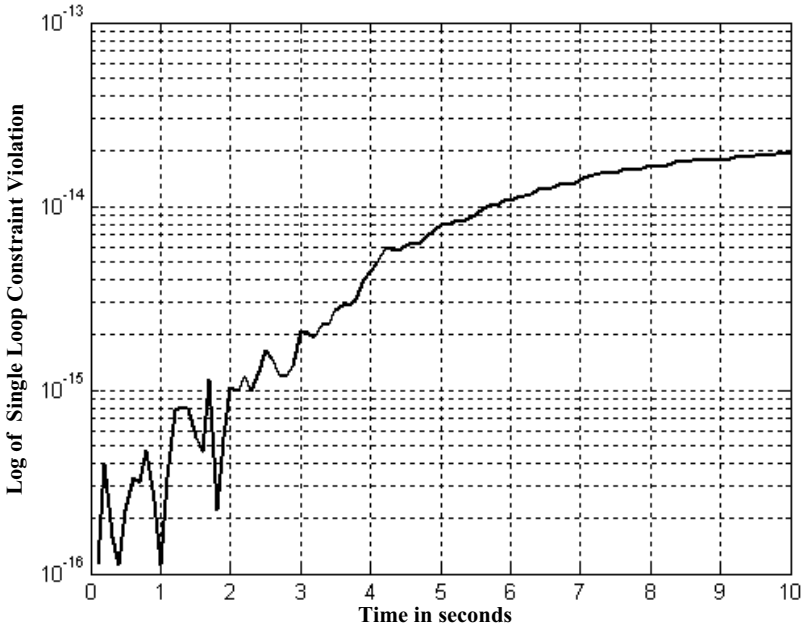


Figure 7: Constraint Violation in Single Loop

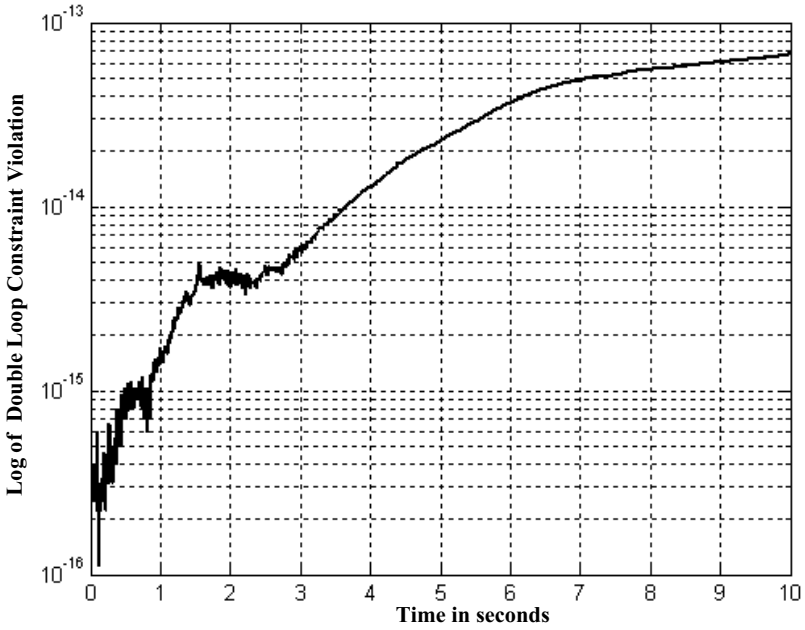
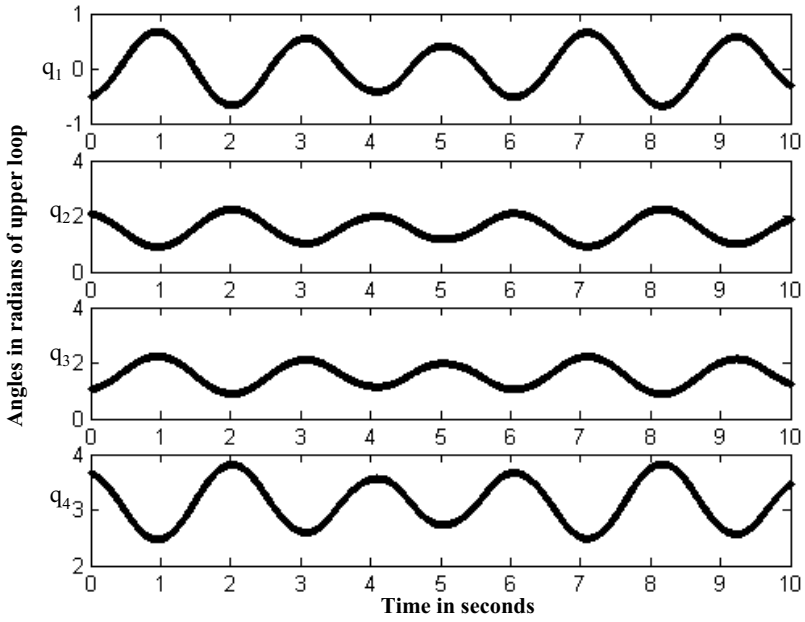


Figure 8: Constraint Violation in Coupled Loops



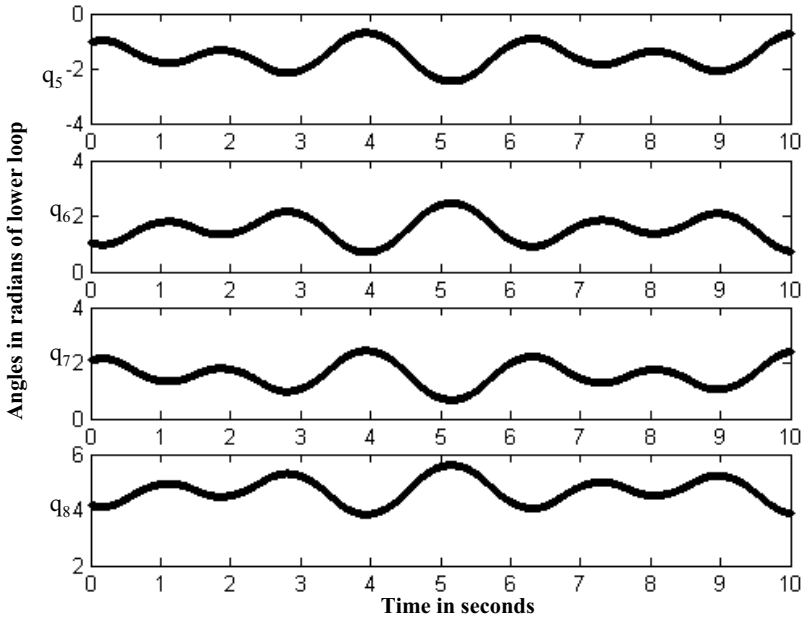


Figure 9: Simulation Results of Multiple Coupled Loops

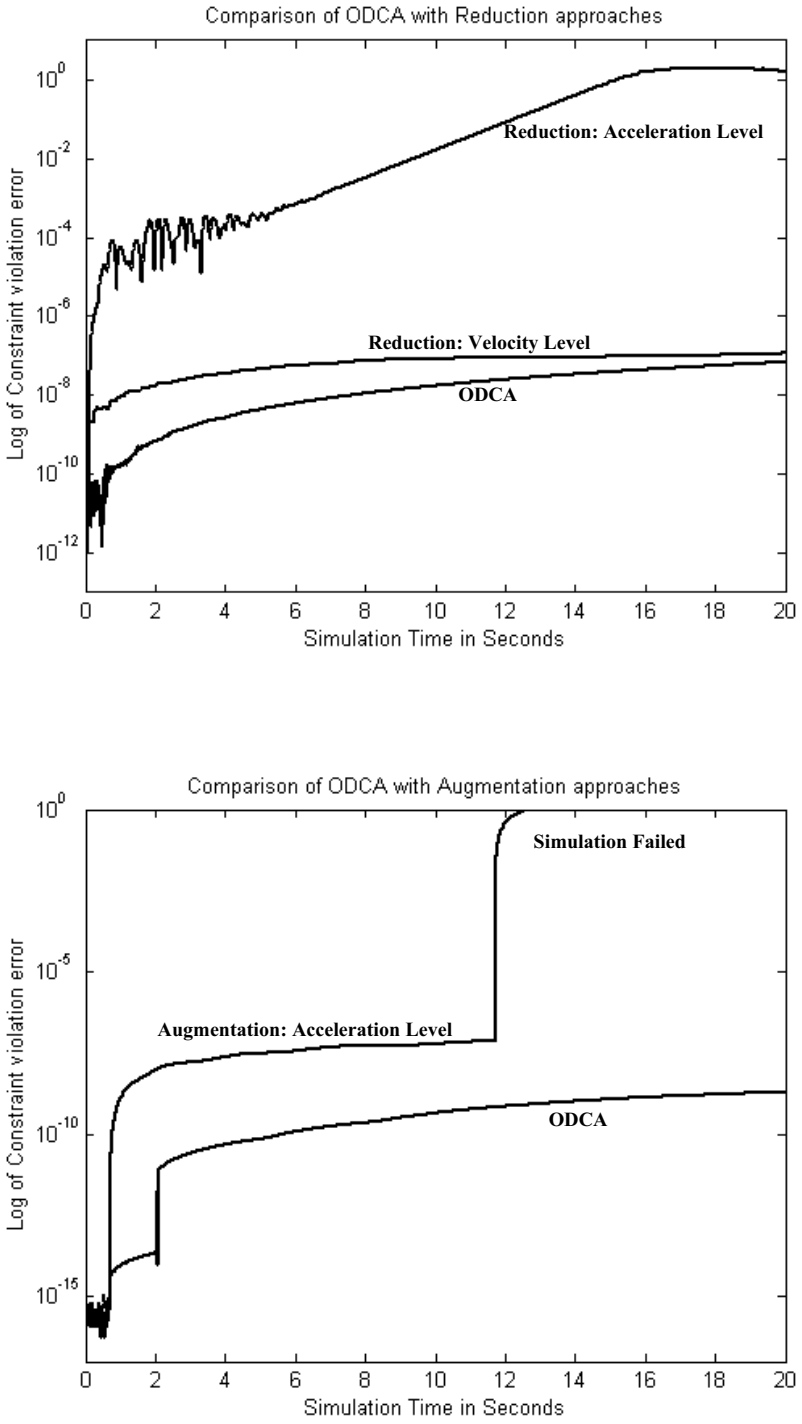


Figure 10: Constraint Violation Error of System with Singular Configuration