Performance Investigation and Constraint Stabilization Approach for the Orthogonal Complement-Based Divide-and-Conquer Algorithm

I.M.Khan*, K.S.Anderson

Computational Dynamics Laboratory,
Department of Mechanical Aerospace and Nuclear Engineering,
Rensselaer Polytechnic Institute, 110 8th St, Troy NY 12180, USA.

Abstract

The introductory paper on the orthogonal complement-based divide-and-conquer algorithm (ODCA) lacked in properly characterizing the growth rate of the constraint violation error and the singularity handling capabilities of the algorithm [1]. In this paper, we investigate the performance of the ODCA with regards to constraint error growth and singularity handling capabilities. Moreover we introduce a new inverse dynamics-based constraint stabilization approach. The proposed method is applicable to general multibody systems with arbitrary number of closed kinematic loops. We compare the ODCA with augmented [2] and reduction [3] methods. Our results indicate that the error growth rate for the ODCA falls between these two traditional techniques. Moreover, using benchmark numerical problems, we illustrate the effectiveness of the stabilization scheme.

Keywords: Constrained Multibody Systems, Orthogonal Complement, Divide-and-Conquer Algorithm, Logarithmic Complexity, Constraint Stabilization

1. Introduction

Mukherjee and Anderson in 2007 presented a new divide-and-conquer-based (DCA-based) algorithm for the dynamic simulations of constrained multibody
systems [1]. The so called orthogonal complement-based divide-and-conquer algorithm (ODCA) provides linear and logarithmic complexity when implemented in serial and parallel, respectively. The ODCA is highly efficient and provides basis for a new class of DCA-based multibody dynamics algorithms. These algorithms include dynamic simulation of articulated flexible bodies [4]; handling discontinuous changes in the model through generalized momentum balance [5, 6]; contact mechanics [7]; sensitivity analysis [8, 9]; and modeling generalized forces as well as control torques [10, 11]. As such, we investigate the ODCA and revisit some of the important aspects of the original ODCA paper. In this work, we review the ODCA as applied to closed loop systems and singular configurations. The constraint violation error growth rate for the ODCA is determined and compared with traditional multibody dynamics techniques. Furthermore, an inverse dynamics-based constraint stabilization technique is presented, which is applicable in the divide-and-conquer architecture.

1.1. Historical Perspective

In 1999, Roy Featherstone presented the divide-and-conquer algorithm (DCA) for the parallel computation of multi-rigid-body dynamics [12, 13]. The DCA is the first truly time-optimal multibody dynamics algorithm that achieved theoretical logarithmic complexity, $O(\log(n))$, on $O(n)$ processors ($n$ being the number of bodies in the system). In 2004, Critchley and Anderson [14] presented an efficient form of the DCA (DCAe) that exploited the recursive nature of the multi-body tree topology. They combined the DCA with Anderson’s $O(n)$ algorithm [15] to boost speed. In 2007, Mukherjee and Anderson extended the basic DCA to general multibody systems with closed kinematic loops. They presented an exact and non-iterative orthogonal complement-based DCA or ODCA [1]. Recently, Malczyk and Frączek proposed an iterative form of the DCA that utilizes the augmented Lagrangian and mass-orthogonal projection approach at the position and velocity level, to prevent growth in constraint violation error [16].

Among different variations of the DCA, the ODCA may be regarded as the most basic and efficient form. The ODCA imposes the constraints on the bodies at the acceleration level. This is achieved through exploiting kinematic relationships involving joint motion subspaces and their orthogonal complements. In order to demonstrate the validity and usefulness of the algorithm, the authors of the original ODCA paper presented several numerical examples and the results were compared with those obtained using other multibody dynamics methods. However, the ODCA paper lacked in certain aspects, such as, correct calculation of
error growth rate and the performance of the algorithm near singular configurations. In this respect, we feel the need to revisit ODCA and present the reader with more insight into the algorithm. Moreover, a constraint stabilization methodology applicable to the divide-and-conquer architecture is also presented.

1.2. Contribution

One of the challenges in the simulations of constrained multibody systems is the growth in the constraint violation error. This accumulation in the error occurs during the numerical integration of the equations of motion. The drifts in the numerical solution can eventually turn into serious constraint violations, causing simulation failure or non-physical results. The quality of the initial conditions, the accuracy and the robustness of the method used for the formation of the equations of motion, and the numerical integrator employed during the simulation are some of the factors that drive the overall behavior of the error. Another important aspect in multibody dynamic simulations involving loop topologies is the presence of geometric singularities. These are the kinematic configurations at which certain system matrices become rank deficient or ill-conditioned (near those regions), causing jumps in the steady error growth and thus, can very quickly lead to simulation failure. In this paper, we show that for the constrained multibody systems the error growth in the ODCA is comparable to the traditional methods. Some commonly known problematic systems will be simulated to compare the growth rate and singularity handling capability of the algorithm. Since the behavior in the error is highly dependent on various factors, therefore, it is necessary to maintain uniformity while comparing distinct simulation techniques. We limit ourselves to compare the ODCA with two techniques for handling constrained multibody systems, namely, augmented method with Lagrange multipliers and reduction through coordinate partitioning and embedding [2,3]. Moreover, a constraint stabilization technique for the ODCA is also presented in this paper. The stabilization technique is based on the inverse dynamics control law and introduces a corrective term in the constraint forces necessary to close the kinematic loop. For very large and complex systems, with many closed kinematic loops, the stabilization technique is easy to implement at almost zero computational cost.

In section 2 a brief overview of the ODCA is presented. The constraint stabilization technique is presented in section 3. Next, numerical test cases are presented before conclusion.
2. Orthogonal Complement-based Divide-and-Conquer Algorithm

The ODCA uses the spatial form the Newton-Euler equations to form the equations of motion of the individual bodies at particular points [17], also termed as handles (the interested reader may consult this work for a more detail description of the basic method). A handle can be any important point with which the body interacts with the environment, but most often, it corresponds to the location of a joint on the body. The equations of motion in the ODCA are written in terms of inverse inertias and bias accelerations at these handles. Although this discussion is limited to bodies possessing two handles for clarity, these may be easily extended to an arbitrary number of body handles. Consider a body \( k \) connected to the other bodies in the multibody tree by two joints at its ends. These joints act as the locations for the two handles on the body and the equations corresponding to each of these two handles, termed the two-handle equations of motion, can be given as

\[
A^k_1 = \zeta^{k}_{11} F^{k}_{1c} + \zeta^{k}_{12} F^{k}_{2c} + \zeta^{k}_{13},
\]

\[
A^k_2 = \zeta^{k}_{21} F^{k}_{1c} + \zeta^{k}_{22} F^{k}_{2c} + \zeta^{k}_{23}.
\]

In the above equations, \( A^k_i \) and \( F^{k}_{ic} \) \((i = 1 \ldots 2)\) represent the spatial acceleration and spatial constraint force, respectively. The inverse inertias at two handles are represented by \( \zeta^{k}_{ij} \) \( (i = 1 \ldots 2, j = 1 \ldots 2) \), where as, \( \zeta^{k}_{i3} \) \( (i = 1 \ldots 2) \) contains the bias accelerations, as well as the effects of all external forces on the body. The equations for body \( k + 1 \) can also be given in the same manner

\[
A^{k+1}_1 = \zeta^{k+1}_{11} F^{k+1}_{1c} + \zeta^{k+1}_{12} F^{k+1}_{2c} + \zeta^{k+1}_{13},
\]

\[
A^{k+1}_2 = \zeta^{k+1}_{21} F^{k+1}_{1c} + \zeta^{k+1}_{22} F^{k+1}_{2c} + \zeta^{k+1}_{23}.
\]

An illustration of bodies \( k \) and \( k + 1 \) is shown in fig. Assuming that the two bodies are connected through a joint \( J^k \), that permits \( \nu_k \) number of degrees of freedom (dof), the kinematic relationship between the two bodies can be given by

\[
A^{k+1}_1 - A^k_2 = P^k \dot{u} + \dot{P}^k u.
\]

In the above equation, \( P^k \) represents a \( 6 \times \nu_k \) matrix that maps the generalized joint relative speeds \( u \) and accelerations \( \dot{u} \) into \( 6 \times 1 \) spatial vector of the corresponding quantities. Let us also introduce \( D^k \) as the orthogonal complement of the matrix \( P^k \). As such, by definition one may write

\[
(D^k)^T P^k = 0_{(6-\nu_k)\times\nu_k}.
\]
For example, in case of a revolute joint which allows rotation about the $z$ axis, $\mathcal{P}^k$ and $\mathcal{D}^k$ can be given as

$$
\mathcal{P}^k = \begin{bmatrix}
0 \\
0 \\
1 \\
0 \\
0 \\
0
\end{bmatrix},
\mathcal{D}^k = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}.
$$

(7)

Note that the orthogonal complement $\mathcal{D}^k$ maps the constraint forces acting at the joints into the spatial constraint force vector. Moreover, at the joint $J^k$ the constraint forces acting on bodies $k$ and $k+1$ are equal and opposite

$$
F_{2c}^k = -F_{1c}^{k+1}.
$$

(8)

The ODCA has two main passes, hierarchic assembly and disassembly. In the assembly process, the two-handle equations of motion for individual bodies are combined to form the two-handle equations for the resultant assembly. For example, using eqns. 5 and 8, the two-handle equations for the assembly $k : k + 1$ can be obtained as

$$
A_1^k = \zeta_{11}^{k:k+1} F_{1c}^k + \zeta_{12}^{k:k+1} F_{2c}^{k+1} + \zeta_{13}^{k:k+1},
$$

(9)

$$
A_2^{k+1} = \zeta_{21}^{k:k+1} F_{1c}^k + \zeta_{22}^{k:k+1} F_{2c}^{k+1} + \zeta_{23}^{k:k+1}.
$$

(10)

At the end of the assembly process, the two-handle equations for a single assembly representing all the bodies in the system are obtained and the disassembly process may begin.

The two-handle equations for the final assembly can be solved using the boundary conditions. Consider a single closed loop consisting of $n$ number of bodies and the ground. The kinematic relationship between the assembly and the ground can be given as

$$
A_1^1 = \mathcal{P}^{b_j} \dot{u}_{b_j} + \dot{\mathcal{P}}^{b_j} \dot{u}_{b_j},
$$

(11)

$$
A_2^n = \mathcal{P}^{t_j} \dot{u}_{t_j} + \dot{\mathcal{P}}^{t_j} \dot{u}_{t_j},
$$

(12)

where subscripts $b_j$ and $t_j$ represents the base and terminal joints, respectively. Equations 11 and 12 can also be written as

$$
\mathcal{P}^{b_j} \dot{u}_{b_j} = \zeta_{11}^{b_j} F_{1c}^1 + \zeta_{12}^{b_j} F_{2c}^n + \zeta_{13}^{b_j},
$$

(13)

$$
\mathcal{P}^{t_j} \dot{u}_{t_j} = \zeta_{21}^{t_j} F_{1c}^1 + \zeta_{22}^{t_j} F_{2c}^n + \zeta_{23}^{t_j}.
$$

(14)
where $\dot{P}_1 u_1$ and $\dot{P}_{n+1} u_{n+1}$ terms are absorbed in the bias acceleration terms. Pre-multiplying each of the above equation by the transpose of the corresponding orthogonal complement matrix gives

$$0 = (D_{bj})^T [\xi_{11} D_{bj} f_{1c} + \xi_{12} D^{ij} f_{2c} + \xi_{13}], \quad (15)$$

$$0 = (D^{ij})^T [\xi_{21} D_{bj} f_{1c} + \xi_{22} D^{ij} f_{2c} + \xi_{23}], \quad (16)$$

where $f_{ic} (i = 1 \ldots 2)$ is the matrix of constraint forces and torques acting on joints and are called constraint force measure numbers [18]. The constraint force measure numbers of a joint $i$ are related to the spatial constraint force vector as

$$F_i = D^i f_i.$$  

After some algebraic manipulations the constraint forces are obtained as

$$\begin{bmatrix} f_{1c}^1 \\ f_{2c}^n \end{bmatrix} = \begin{bmatrix} \chi_{11} & \chi_{12} \\ \chi_{21} & \chi_{22} \end{bmatrix}^{-1} \begin{bmatrix} \chi_{13} \\ \chi_{23} \end{bmatrix}, \quad (18)$$

where

$$\chi \equiv \begin{bmatrix} \chi_{11} & \chi_{12} \\ \chi_{21} & \chi_{22} \end{bmatrix} = \begin{bmatrix} (D_{bj})^T [\xi_{11} D_{bj} + \xi_{12} D^{ij}] & (D^{ij})^T [\xi_{21} D_{bj} + \xi_{22} D^{ij}] \end{bmatrix}, \quad (19)$$

is the matrix associated with the mass and inertia terms of the final system assembly and

$$\begin{bmatrix} \chi_{13} \\ \chi_{23} \end{bmatrix} = \begin{bmatrix} (D_{bj})^T [\xi_{13}] \\ (D^{ij})^T [\xi_{23}] \end{bmatrix}, \quad (20)$$

is the matrix associated with state dependent terms and applied forces.

Using the constraint forces of eqn. [18], the accelerations for the two handles of the root assembly can be easily obtained. These accelerations and constraint forces serve as the boundary conditions for the higher level assemblies. At the end of the disassembly process the constraint forces and the accelerations for all the bodies in the system are obtained. An illustration of the assembly and disassembly process is shown in fig. 2.

### 3. Constraint Error Stabilization

In the absence of any stabilization technique, the small errors, which may occur in the initial conditions or which may arise naturally during the numerical
integration, are bound to amplify over the course of the simulation. As such, a simple yet effective technique within the DCA framework is presented to deal with the numerical drift. The error stabilization technique is based on the inverse dynamics control law and is applicable to any system with arbitrary number of constraints and loops. The stabilization technique works in a similar manner as the Baumgarte’s method [19], i.e. penalize the system based on the position and velocity constraint violation error. However, the Baumgarte’s stabilization method
requires the explicit formulation of the acceleration level constraint equations, whereas the present method works in conjunction with the ODCA scheme and adds a penalty force to the constraint forces at the loop closure joint.

Consider the two-handle equations of motion for the loop assembly consisting bodies 1 to \( n \)

\[
A_1^1 = \zeta_{11}^1 F_{1c}^1 + \zeta_{12}^1 F_{2c}^1 + \zeta_{13}^1 , \quad (21)
\]

\[
A_2^1 = \zeta_{21}^1 F_{1c}^1 + \zeta_{22}^1 F_{2c}^1 + \zeta_{23}^1 . \quad (22)
\]

Since the ODCA uses an internal set of coordinates, therefore the constraints at the internal joints are satisfied exactly. We assume that the kinematic constraint equations are defined for handle-2, which due to numerical inconsistencies are not fully satisfied. In order to remove this error, the spatial constraint force acting at handle-2 is changed to \( \bar{F}_{2c}^n \) by adding a corrective term, \( R_2 \). The modified spatial acceleration, \( \bar{A}_2^n \) at handle-2 can be given as

\[
\bar{A}_2^n = \zeta_{21}^1 F_{1c}^1 + \zeta_{22}^1 \bar{F}_{2c}^n + \zeta_{23}^1 , \quad (23)
\]

where

\[
\bar{F}_{2c}^n = F_{2c}^n + R_2 . \quad (24)
\]

In eqn. \( 24 \), \( R_2 \) is the control input force at handle-2. Setting \( R_2 \) as

\[
R_2 = [\zeta_{22}^1]^{-1} [\ddot{X}_2 - \zeta_{21}^1 F_{1c}^1 - \zeta_{22}^1 F_{2c}^n - \zeta_{23}^1] , \quad (25)
\]

and substituting eqn. \( 25 \) in eqn. \( 23 \) results in

\[
\bar{A}_2^n = \dddot{X}_2 . \quad (26)
\]

In eqn. \( 26 \), \( \dddot{X}_2 \) is an unknown control parameter which is yet to be chosen. Let us choose \( \dddot{X}_2 \) as,

\[
\dddot{X}_2 = A_2^n + K_D \dot{e} + K_P e , \quad (27)
\]

where \( K_D \) and \( K_P \) are \( 6 \times 6 \) matrices of gains associated with the inverse dynamics control law \[20\]. The terms \( e \) and \( \dot{e} \) are the spatial position and velocity errors, respectively. These error vectors are given as

\[
e = D^n (q - \bar{q}) , \quad (28)
\]

\[
\dot{e} = D^n (\dot{q} - \dddot{\bar{q}}) . \quad (29)
\]
In the above equations, $q$ and $\dot{q}$ are the actual positions and velocities at handle-2, $\bar{q}$ and $\bar{\dot{q}}$ are the desired positions and velocities at handle-2, and $D^n$ is the orthogonal complement matrix of loop closure joint $n$. Combining equations eqns. (26) and (27) results in

$$\ddot{e} + K_D \dot{e} + K_P e = 0.$$  \hspace{1cm} (30)

Equation (30) governs the error dynamics of the system. Substituting (27) in (25) gives

$$R_2 = [\zeta_{22}^{1:n}]^{-1} [A^n_2 + K_D \dot{e} + K_P e - \zeta_{21}^{1:n} F_{1c} - \zeta_{22}^{1:n} F_{2c} - \zeta_{23}^{1:n}].$$  \hspace{1cm} (31)

Finally, using the definition of $A^n_i$ from eqn. (22), the expression for the control force input is obtained as

$$R_2 = [\zeta_{22}^{1:n}]^{-1} [K_D \dot{e} + K_P e].$$  \hspace{1cm} (32)

The new top level assembly equations can now be represented as

$$\bar{A}_1^1 = \zeta_{11}^{1:n} F_{1c} + \zeta_{12}^{1:n} F_{2c} + \zeta_{13}^{1:n},$$  \hspace{1cm} (33)

$$\bar{A}_2^n = \zeta_{21}^{1:n} F_{1c} + \zeta_{22}^{1:n} F_{2c} + \zeta_{23}^{1:n}.$$  \hspace{1cm} (34)

The stabilization process works by calculating the corrective forces $R_2$ at each time step using eqn. (32). For example, at time step $i$, the position and velocity errors ($e$ and $\dot{e}$) can be calculated using the actual ($q$ and $\dot{q}$) and desired ($\bar{q}$ and $\bar{\dot{q}}$) values of positions and velocities. Using these errors and control gains the stabilization forces can be calculated for each individual closed kinematic loop in the system. After the disassembly process the accelerations $\bar{A}_1^1$ and $\bar{A}_2^n$ are obtained and integrated to find the actual positions and velocities for the next time step $i + 1$. An illustration of the constraint stabilization scheme is shown in fig. 3.

4. Numerical Examples

In this section, the error growth rate for the ODCA, effects of singularities and the performance of the constraint stabilization scheme is presented. Since there exist a number of factors such as initial conditions, type of numerical integrator and size of time step, that may influence a numerical simulations, therefore, care is taken to maintain uniformity among various code. Unless otherwise specified, fourth-order Runge-Kutta (RK4) time integrator is employed in all cases for numerical integration. Identical initial conditions and equal time steps are used for simulations that compare different multibody methods.
4.1. Error Growth Rate

With the traditional approaches, the error grows in a quadratic fashion when the constraints are imposed at the acceleration level. If the constraints are im-
posed at both acceleration and velocity levels, the error has been reported to grow linearly with time \[21, 22\]. In \[1\], the growth rate of the error for the ODCA was claimed to be \(t^{0.7}\), which is significantly better than the augmented and reduction approach. Therefore, we study the growth rate for the ODCA using a planar four bar deltoid mechanism, as shown in fig. 4(a). Equation (7) defines the joint motion subspace and orthogonal complement for the revolute joints in the system. The mass, length, and inertia of each bar is taken as unity and the initial value of the independent dof \((\theta_1)\) is \(-\frac{\pi}{6}\) rads. The system is started from rest under the influence of gravity and simulated for 40 s. The plot of the angular position obtained by the ODCA and the \(L^2\) norm of the loop closure error is shown in fig. 5. The results indicate that the growth rate of the ODCA falls between the augmented and the reduction approach.

![Graph of angular position and \(L^2\) norm of error](image)

Figure 5: Plots of, a) angular position, and b) \(L^2\) norm of the error.

An extended simulation of 2000 s is performed to properly determine and compare the error growth rates as a function of time. The \(L^2\) norms of the error from different methods are normalized w.r.t to the initial error value. The normalized error is then fitted with \(t^\gamma\), where \(\gamma\) is the unknown constant to be determined. The behavior of the error growth rate was found to be varying over the course of the simulation. As such, the exponential fits are performed on four different segments of the transient error response. The values of \(\gamma\) are shown in tab. 1.

It may be noticed that reduction or coordinate embedding approach performed better than the other two methods. The ODCA performed better than the augmented method, with which the constraint violation error grows almost quadrat-
Table 1: Exponent of fit ($t^\gamma$) for the error growth rate

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>DCA</th>
<th>Augmented</th>
<th>Embedding</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-500</td>
<td>1.93</td>
<td>2.00</td>
<td>1.78</td>
</tr>
<tr>
<td>500-1000</td>
<td>1.87</td>
<td>1.96</td>
<td>1.83</td>
</tr>
<tr>
<td>1000-1500</td>
<td>1.83</td>
<td>1.94</td>
<td>1.83</td>
</tr>
<tr>
<td>1500-2000</td>
<td>1.82</td>
<td>1.91</td>
<td>1.80</td>
</tr>
</tbody>
</table>

This behavior can be explained through the differences in the constraints imposition practice in the three methods. In the constraint embedding method, the independent dof are identified and a minimum $(n - m)$ number of the ODEs are formed ($n$ being the number of dof with $m$ number of constraint equations). The remaining dependent coordinates are found using the acceleration level constraint equations, thereby, greatly reducing the constraint violation error. One could also use the same method to eliminate the redundant velocity variables to further reduce the constraint violation error [3, 23]. The augmented approach uses the redundant set of coordinates and enforces the constraints via Lagrange multipliers, which are calculated along with the accelerations, by inverting a larger $(n + m) \times (n + m)$ matrix [2, 24]. In contrast, although the ODCA carries forward redundant set of generalized coordinates, it imposes the constraints at the individual joints exactly at the acceleration level through joint motion maps. The error growth in the ODCA is also affected by a larger number of matrix multiplications and inversions during the assembly and disassembly process.

The constraint enforcement method in the embedding technique, which eliminates the redundant variables, clearly marks a greater level of accuracy as compared to the ODCA and augmented methods. Since the ODCA enforces the constraints exactly at the acceleration level at the joints therefore it shows better accuracy than the augmented approach. However, it may be noted that as opposed to the claims of the original ODCA paper, the growth rates of all three methods are comparable with small differences. Even though the behavior of the error is influenced by many factors, such as, type of the problem, size of the time step, and integrator, it can be inferred from the numerical test results and the above discussion that the error growth rate of the ODCA falls between the augmented and reduction approach.

4.2. Handling Singularities

Presence of kinematic singularities is a widely known issue in the multibody dynamics simulations. These singularities are the geometric configurations where
the constraint Jacobian looses rank or the number of independent constraint equations reduces \([25, 26]\). Various methods have been proposed in literature to deal with singularities in the multibody dynamics simulations \([27, 28, 29, 30]\). The ODCA claimed to overcome the singularity problem through the use of redundant set of generalized coordinates and the absence of the constraint Jacobian \([1]\). However, it may be noted that the kinematic constraints are embedded into the ODCA formulation during the assembly process. Therefore, it is necessary to revisit and review the singularity handling capabilities of the ODCA.

As shown in fig. 4, two different mechanisms are used to demonstrate the performance of the ODCA in the presence of singularities. Once again the physical parameters are taken as unity and the initial value of \(\theta_1\) is \(\frac{\pi}{6}\) rads. The time step for the RK4 integrator is fixed at 0.01 s and both systems are simulated for 10 s. Figure 6 shows the plot of the error for the two mechanisms using the three approaches. It can be noticed that in both cases the ODCA performed similar to the augmented and embedding approach.

The deltoid suffers from multiple singularities at the configurations when \(\theta_1\) is 0 or \(-\pi\) rads. Similarly, the four bar mechanism passes through the singularity when \(\theta_1\) is 0 rads. Note that the singularities experienced by these two mechanisms are different in nature. For the deltoid, the constraint Jacobian becomes rank deficient at singular positions, therefore the mass matrix or the system matrix (which is inverted at each time step, \(\chi\) is case of the ODCA) becomes ill-conditioned. By comparison, only the dependent partition of the constraint Jacobian becomes rank deficient in the four bar mechanism, thus causing error in the forcing terms. The system matrix remains well conditioned at the singular positions for the four bar mechanism. The singularities defined for the deltoid and the four bar can be classified as Type II and Type III, respectively, according to \([31]\).

In case the deltoid mechanism, significant jumps in the error can be observed in fig. 6(a). Even though the ODCA does not carry the constraint Jacobian explicitly, but the inverted \(\chi\) matrix given in eqn. \([18]\) becomes ill-conditioned in the vicinity of singular positions. The four bar mechanism is handled well by all three approaches, as shown in fig. 6(b). Figure 7 shows the condition number of the \(\chi\) matrix for the two mechanisms. The results indicate that the ODCA performs similar to the traditional approaches when dealing with singular configurations and may require special handing of the constraints.

### 4.3. Constraint Stabilization

In case of singularities, the DCA based stabilization technique can not fix the rank deficient \(\chi\) matrix, but it can act as means to keep the error under control. The
stabilization technique works on a similar principle as the Baumgarte’s method and requires proper selection of the control gains $K_D$ and $K_P$. The Baumgarte’s parameters are infamous for the difficulty associated with their choice and often requires specialized methods for their determination [26, 32]. An incorrect choice of these parameters may lead to instabilities and simulation failure. Although the DCA based scheme is different from Baumgarte’s method, but its control parame-

Figure 6: Plots of $L^2$ norm of the errors.

Figure 7: Plot of the condition number of the system matrix ‘χ’.
ters can be estimated using similar approaches that are suggested for Baumgarte’s method [33, 34]. Here we have selected a simple criteria for the selection of the control parameters described in [35]. Let \( h \) be the time step then the control gains \( K_D \) and \( K_P \) are estimated using

\[
K_D = \frac{2}{h},
\]

(35)

\[
K_P = \frac{K_D^2}{4}.
\]

(36)

In eqn. 36, the relationship between \( K_D \) and \( K_P \) ensures asymptotical stability and critical damping for eqn. 30 which governs the error dynamics of the system. It may be noted that apart from the above mentioned technique there exist infinite set of \( K_D \) and \( K_P \) that one can choose. However, if the values of these parameters are too small they may not be able to effectively control the constraint violation error. On the other hand, large values may cause the system to exhibit stiff behavior resulting in instability.

The deltoid mechanism as shown in fig. 4(a), and the Andrew’s squeezers mechanism as shown in 9 are selected to test the stabilization technique.

4.3.1. Deltoid

The deltoid mechanism is simulated for the duration of 10 s, with the stabilization in effect. The physical parameters and the initial condition are kept same as the singular case of section 4.2; the time step \( h \) is taken as 0.01 s. Figure 8 shows the plots of the angular position and error growth with and without the stabilization. The results indicated that when no stabilization is used the error quickly grows to very large value, causing the simulation failure. When the constraint stabilization is enforced, the system is able to recover from the jumps in the error due to singularities. The stabilization scheme is able to keep the error under control and close to machine accuracy.

4.3.2. Andrew’s Squeezers Mechanism

Andrew’s squeezers mechanism is considered as the second test case. This mechanism has been studied by numerous authors and it serves as a benchmark problem to test multibody codes [36, 37, 38, 39, 40]. The system is composed of seven bodies and a stiff spring, as shown in fig. 9. There are three kinematic loops in the system and the mechanism is driven from rest by an applied torque on body 1. Complete description of the mechanism including the initial conditions is taken from [41]. The system is simulated using MATLAB® ODE45 integrator for 0.03
s with output at each $5e^{-4}$ s. The absolute and relative tolerances are set to $1e^{-6}$.

Since a variable step integrator is used that may take very small time steps during the numerical simulation, therefore in order to avoid instability, the value of $h$ is fixed to $5e^{-4}$ s. This value is found by trial and error and gives good results. The corrective forces for the three kinematic loops are calculated during the disassembly process and applied to each loop closure joint. The plots of the angles and
errors are shown in fig. [10] The plots show that the stabilization scheme works well and the results are in good agreement with the published solutions.

5. Conclusion

Investigation of the ODCA reveals that incorrect conclusions were drawn in the introductory ODCA paper [1]. Using several numerical examples, the error growth rate for the ODCA is quantified. It is shown that the error growth rate for the ODCA is comparable to other multibody methods that impose constraints at the acceleration level. Moreover, it is demonstrated that the ODCA suffers from poor conditioning issues near singular configurations. A new constraint stabili-
lization scheme for the ODCA is presented and tested on two problematic mechanisms. The constraint violation error response indicates that the new method works well in conjunction with the ODCA.

Acknowledgement

The authors would like to acknowledge the financial support of the Division of Civil, Mechanical & Manufacturing Innovation of the National Science Foundation (grant # 0757936 and 1161872). The authors would also like to thank Dr. Muhammad Poursina and Jeremy Laflin for their help and insight into the problem.

References


Figure captions and Tables

Figure 1: Individual bodies connected by kinematic joint.

Figure 2: The hierarchic assembly-disassembly process in the DCA.

Figure 3: Illustration of the constraint stabilization control loop.

Figure 4: a) Four bar deltoid mechanism, and b) four bar mechanism.

Figure 5: Plots of, a) angular position, and b) $L^2$ norm of the error.

Figure 6: Plots of $L^2$ norm of the errors.

Figure 7: Plot of the condition number of the system matrix ‘$\chi$’.

Figure 8: Plots of, a) angular position, and b) $L^2$ norm of the error.

Figure 9: Illustration of Andrews squeezers mechanism.

Figure 10: Plots of the angles and $L^2$ norm of the errors for the Andrew’s squeezer mechanism.
Table 1: Exponent of fit ($t^\gamma$) for the error growth rate

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>DCA</th>
<th>Augmented</th>
<th>Embedding</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-500</td>
<td>1.93</td>
<td>2.00</td>
<td>1.78</td>
</tr>
<tr>
<td>500-1000</td>
<td>1.87</td>
<td>1.96</td>
<td>1.83</td>
</tr>
<tr>
<td>1000-1500</td>
<td>1.83</td>
<td>1.94</td>
<td>1.83</td>
</tr>
<tr>
<td>1500-2000</td>
<td>1.82</td>
<td>1.91</td>
<td>1.80</td>
</tr>
</tbody>
</table>
Figure 1: Individual bodies connected by kinematic joint.

Figure 2: The hierarchic assembly-disassembly process in the DCA.
Figure 3: Illustration of the constraint stabilization control loop.

Figure 4: a) Four bar deltoid mechanism, and b) four bar mechanism.
Figure 5: Plots of, a) angular position, and b) $L^2$ norm of the error.

Figure 6: Plots of $L^2$ norm of the errors.
Figure 7: Plot of the condition number of the system matrix \( \chi \).

Figure 8: Plots of, a) angular position against time, and b) \( L^2 \) norm of the error.
Figure 9: Illustration of Andrew’s squeezers mechanism.
Figure 10: Plots of the angles and $L^2$ norm of the errors for the Andrew’s squeezer mechanism.