

Flight Dynamic Simulation for Multibody Aircraft Configurations

Emily Leylek,^{*} Michael Ward,[†] and Mark Costello[‡]
Georgia Institute of Technology, Atlanta, Georgia 30332

DOI: 10.2514/1.55858

To capture salient features of the flight dynamic behavior of some aircraft configurations, the air vehicle is best idealized as a multibody dynamic system where individual rigid bodies are connected via various joint connections. This paper considers automatic generation of a multibody air vehicle simulation where the standard six degrees of freedom rigid body modeling forms the basic kernel of the method. By adding an extra term to the individual body equations of motion representing joint connection constraint loads, a multibody flight dynamic model can be constructed. A globally stable nonlinear controller computes these constraint forces and moments. Through three modeling examples (projectile with an internal moving mass, airdrop system, and articulated wing aircraft), practical aspects of the methodology are explored. Many nonlinear control techniques can be used for the constraint control problem. Here, feedback linearization and sliding mode control algorithms are shown to work effectively. Constraint computation dominates overall simulation time, independently of the control algorithm used, and several means to reduce simulation time are reported. Moreover, unconstrained linear dynamic models can be generated by combining a linear model of the constrained coordinate equations of motion and a linear model of the constraint equations.

Nomenclature

E_T, E_R	= translational, m; rotational error components, rad
E_T^*, E_R^*	= translational, m; rotational error components selected for specific joint, rad
F	= unconstrained dynamic equations
$\mathbf{F}_C, \mathbf{M}_C$	= joint constraint force, N; moment, N · m
F_X, F_Y, F_Z	= force measure numbers in body frame, N
F_{TK}, F_{RK}	= translational, rotational kinematic equations
F_{TD}, F_{RD}	= translational, rotational dynamic equations
G	= constraint force, moment transformation matrix
I	= mass moment of inertia matrix, kg · m ²
$\mathbf{I}_B, \mathbf{J}_B, \mathbf{K}_B$	= basis vectors for body reference frame
$\mathbf{I}_C, \mathbf{J}_C, \mathbf{K}_C$	= basis vectors for child body reference frame
$\mathbf{I}_I, \mathbf{J}_I, \mathbf{K}_I$	= basis vectors for inertial reference frame
$\mathbf{I}_P, \mathbf{J}_P, \mathbf{K}_P$	= basis vectors for parent body reference frame
i	= subscript for i th rigid body
j	= subscript for j th joint
m	= mass [kg]
M_X, M_Y, M_Z	= moment measure numbers in body frame, N · m
q_0, q_1, q_2, q_3	= quaternion orientation parameters, nd
$\mathbf{r}_{O \rightarrow \oplus}$	= position vector from inertial reference to center of mass, m
$S_{\omega_P}, S_{\omega_C}$	= skew symmetric cross product operator acting on angular rates of parent, child body

S_P, S_C	= cross product operator acting on position from mass center of parent, child body to joint location
T_B	= transformation from inertial to body reference frame
T_P, T_C	= transformation from inertial to parent, child reference frame
T_{PJ}, T_{CJ}	= transformation from parent, child to parent joint, child joint reference frame
u, v, w	= velocity vector scalar numbers in body reference frame, m/s
U	= constraint force, N; moment vector, N · m
$\mathbf{v}_{\oplus/I}$	= velocity vector for center of mass with respect to inertial reference frame, m/s
X	= state vector
x, y, z	= position vector measure numbers in inertial reference frame, m
$\Delta X_P, \Delta Y_P, \Delta Z_P$	= vector from parent mass center to parent joint location expressed in parent reference frame, m
$\Delta X_C, \Delta Y_C, \Delta Z_C$	= vector from child mass center to child joint location expressed in child reference frame, m
γ	= pseudo control
Γ_T, Γ_R	= translational, rotational error component selection matrix
ω_n	= natural frequency, rad/s
ξ	= damping ratio [nd]

I. Introduction

ALTHOUGH rigid six degrees of freedom representations are far and away the most common for air vehicle flight dynamic modeling and analysis, there are circumstances where more sophisticated system topology is required. In many instances, the vehicle is idealized with multiple rigid bodies connected together such that relative motion takes place between major components that constitute the system. Projectiles and missiles provide some good examples of this situation. For missiles controlled by an internal moving mass, control is generated by moving the location of a mass within a cavity of the body [1–4]. A multicomponent projectile configuration is the dual spin projectile [5,6] where the forward and aft parts of the projectile are attached via a bearing that permits relative roll rotation between the two sections. A gimbal nose

Presented as Paper 7923 at the AIAA Modeling and Simulation Conference, Toronto, Canada, August 2–5, 2010; received 4 August 2011; revision received 23 February 2012; accepted for publication 1 March 2012. Copyright © 2012 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. Copies of this paper may be made for personal or internal use, on condition that the copier pay the \$10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0731-5090/12 and \$10.00 in correspondence with the CCC.

^{*}Graduate Research Assistant, School of Aerospace Engineering. Student Member AIAA.

[†]Graduate Research Assistant, School of Aerospace Engineering. Student Member AIAA.

[‡]Sikorsky Associate Professor, School of Aerospace Engineering. Associate Fellow AIAA.

projectile is configured with the nose mounted on a spherical joint such that it can rotate relative to the main body to reduce aerodynamic jump [7]. Airdrop systems also provide numerous examples where multibody flight dynamic representations are employed. For parafoil and payload aircraft configurations, the parafoil canopy is often connected to the payload through a confluence point such that the canopy and payload rotate relative to one another, leading to modeling the connection with a hinge or spherical joint [8–10]. Multiple round parachutes connected to a single payload are used for airdrop of heavy loads and are dynamically modeled as rigid canopies connected to a payload via a spherical joint. Currently, new air vehicle concepts are being developed that are fundamentally multibody configurations. Articulated micro air vehicles permit the main lifting surfaces of the aircraft to rotate about a hinge with the purpose of alleviating gust response [11]. In all of these cases, the basic dynamic and control response of the air vehicle is driven by relative motion between various components of the system each with substantial mass and inertia. Hence, flight dynamic modeling of these systems requires a more sophisticated representation than permitted by a single rigid aircraft representation.

Forming the dynamic equations of motion for a multibody dynamic system is much more complex because of the existence of internal constraint forces and moments that connect the system together [12–14]. There has been a great deal of work focused on generating dynamic equations of motion for multibody dynamic systems. See [15,16] and the work cited within for a reasonably complete survey on methods and solution techniques. Although many detailed formulations are documented [17–21], the techniques roughly split along two lines pertaining to the manner in which motion is described (constrained or unconstrained coordinates) and how the fundamental dynamic equations are constructed (Newton–Euler or analytical mechanics approach). If unconstrained coordinates are employed, the final equations of motion tend to be quite involved and tedious to construct. On the other hand, if constrained coordinates are used, the resulting equations of motion are differential-algebraic equations, which are challenging to solve [22,23]. When using a Newton–Euler approach to form the equations of motion, each rigid body is examined individually with constraint forces and moments appearing directly in the formulation [24]. On the other hand, analytical mechanics methods such as Lagrange’s or Kane’s equations consider the system as a whole and enable constraint forces and moments to be automatically eliminated from the resulting equations of motion [25,26].

The work reported here is focused on flight dynamic simulation of multibody air vehicles with a formulation in the same spirit as Chiou and Wu [17]. The method treats each rigid body of the aircraft as a general rigid body with six DOF. All constraint forces and moments that arise from system connections (hinge joints, sliders, etc.) are treated as external loads that are equal and opposite on connecting bodies. A multibody dynamic simulation can be constructed by the simple addition of appropriate constraint forces and moments to the rigid six DOF model along with the addition of a system glue code controller, which is a function of the connectivity of the multibody system. The glue code controller generates the constraining forces and moments. The method has the advantage of leveraging rigid six DOF dynamic modeling, a well known technique within the flight dynamic community. The paper develops this overall multibody flight mechanics simulation method and employs it in the simulation of several example systems to show its strengths and weaknesses.

II. Multibody Flight Dynamic Equations of Motion

A generic multibody vehicle is modeled as a collection of N rigid bodies connected by a set of M joints. The equations of motion for the system are first generated by considering each rigid body of the system individually as if it were an isolated body with no connections to other bodies. At this point, all constraint forces and moments that arise from system connectivity are treated as externally applied forces and moments. The constraint forces and moments are equal and opposite on connected bodies.

A. Topology Description

Each of the N bodies has a body fixed coordinate system located at its center of mass, which is related to the inertial reference frame via Eq. (1).

$$\begin{Bmatrix} \mathbf{I}_{B_i} \\ \mathbf{J}_{B_i} \\ \mathbf{K}_{B_i} \end{Bmatrix} = T_{B_i} \begin{Bmatrix} \mathbf{I}_I \\ \mathbf{J}_I \\ \mathbf{K}_I \end{Bmatrix} \quad (1)$$

Each of the M joints joins two of the N bodies together. For each joint, these two bodies are referred to as the parent and child bodies. Every parent and child has its own associated reference frame, referred to as the parent and child frames. When the i th body is assigned the parent role for the j th joint, the parent reference frame is equivalent to its body fixed reference frame. When the i th body is assigned the child role for the j th joint, the child reference frame is equivalent to its body fixed reference frame. The transformations are defined as follows with respect to the inertial frame for the j th joint.

$$\begin{Bmatrix} \mathbf{I}_{P_j} \\ \mathbf{J}_{P_j} \\ \mathbf{K}_{P_j} \end{Bmatrix} = T_{P_j} \begin{Bmatrix} \mathbf{I}_I \\ \mathbf{J}_I \\ \mathbf{K}_I \end{Bmatrix} \quad \begin{Bmatrix} \mathbf{I}_{C_j} \\ \mathbf{J}_{C_j} \\ \mathbf{K}_{C_j} \end{Bmatrix} = T_{C_j} \begin{Bmatrix} \mathbf{I}_I \\ \mathbf{J}_I \\ \mathbf{K}_I \end{Bmatrix} \quad (2)$$

Each joint also has a joint reference frame aligned with the joint connection on the parent and child bodies. These reference frames are referred to as the parent joint and child joint and are defined as follows.

$$\begin{Bmatrix} \mathbf{I}_{PJ_j} \\ \mathbf{J}_{PJ_j} \\ \mathbf{K}_{PJ_j} \end{Bmatrix} = T_{PJ_j} \begin{Bmatrix} \mathbf{I}_{P_j} \\ \mathbf{J}_{P_j} \\ \mathbf{K}_{P_j} \end{Bmatrix} \quad \begin{Bmatrix} \mathbf{I}_{CJ_j} \\ \mathbf{J}_{CJ_j} \\ \mathbf{K}_{CJ_j} \end{Bmatrix} = T_{CJ_j} \begin{Bmatrix} \mathbf{I}_{C_j} \\ \mathbf{J}_{C_j} \\ \mathbf{K}_{C_j} \end{Bmatrix} \quad (3)$$

Figure 1 summarizes the different parent and child joint reference frames.

B. Single-Body Equations of Motion

For the i th rigid body of the system, the dynamic equations of motion can be written in the following affine form.

$$\dot{\mathbf{X}}_i = \mathbf{F}_i + \mathbf{G}_i \mathbf{U} \quad (4)$$

In the above equation, \mathbf{X}_i is the state vector of the i th rigid body of the system. \mathbf{F}_i represents the unconstrained dynamic equations of the i th rigid body, and $\mathbf{G}_i \mathbf{U}$ represents the contributions of the connection constraint forces and moments to the dynamic equations. The vector \mathbf{U} contains the scalar force and moment constraint elements for *all* the connections for the system concatenated into one vector. Because \mathbf{F}_i determines only the unconstrained motion of the body, acceleration terms determined by the movement of the connected bodies, such as Coriolis and centripetal accelerations, enter into the dynamic equations through the joint connection term $\mathbf{G}_i \mathbf{U}$. The details of the composition of this term are given in the next section.

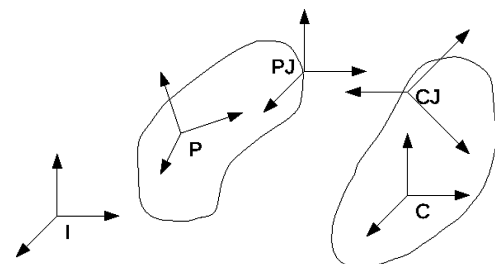


Fig. 1 Two general rigid bodies detailing their associated reference frames.

Each of the N rigid bodies possesses six DOF, including both position and orientation of the body. Equation (5) defines the position vector from the origin of the inertial reference frame to the mass center of the i th rigid body.

$$\mathbf{r}_{O \rightarrow \oplus_i} = x_i \mathbf{I}_I + y_i \mathbf{J}_I + z_i \mathbf{K}_I \quad (5)$$

The orientation of the i th body relative to the inertial reference frame is provided by quaternion rotation parameters [27].

$$\begin{Bmatrix} \mathbf{I}_{B_i} \\ \mathbf{J}_{B_i} \\ \mathbf{K}_{B_i} \end{Bmatrix} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \begin{Bmatrix} \mathbf{I}_I \\ \mathbf{J}_I \\ \mathbf{K}_I \end{Bmatrix} = T_{B_i} \begin{Bmatrix} \mathbf{I}_I \\ \mathbf{J}_I \\ \mathbf{K}_I \end{Bmatrix} \quad (6)$$

In the above equation, q_0 , q_1 , q_2 , and q_3 are the quaternion orientation parameters. The transformation between Euler angles and quaternion orientation parameters is defined in [27]. The translational velocity of the mass center of body i related to the inertial reference frame is described in body coordinates.

$$\mathbf{v}_{\oplus_i/I} = u_i \mathbf{I}_{B_i} + v_i \mathbf{J}_{B_i} + w_i \mathbf{K}_{B_i} \quad (7)$$

The angular velocity of the i th body with respect to the inertial reference frame is also described in body coordinates.

$$\boldsymbol{\omega}_{B_i/I} = p_i \mathbf{I}_{B_i} + q_i \mathbf{J}_{B_i} + r_i \mathbf{K}_{B_i} \quad (8)$$

With these definitions, the state vector for the i th rigid body is defined as follows.

$$\mathbf{X}_i = [x_i \ y_i \ z_i \ q_0 \ q_1 \ q_2 \ q_3 \ u_i \ v_i \ w_i \ p_i \ q_i \ r_i]^T \quad (9)$$

The unconstrained single body equations of motion, F_i , are divided into four contributing elements: translational kinematics, rotational kinematics, translational dynamics, and rotational dynamics

$$\mathbf{F}_i = \begin{Bmatrix} F_{TK_i} \\ F_{RK_i} \\ F_{TD_i} \\ F_{RD_i} \end{Bmatrix} \quad (10)$$

where

$$F_{TK_i} = T_{B_i} \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} \quad (11)$$

$$F_{RK_i} = \frac{1}{2} \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \begin{Bmatrix} p_i \\ q_i \\ r_i \end{Bmatrix} \quad (12)$$

$$F_{TD_i} = \frac{1}{2} S_{\omega_i} \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} + \frac{1}{m_i} \begin{Bmatrix} F_{X_i} \\ F_{Y_i} \\ F_{Z_i} \end{Bmatrix} \quad (13)$$

$$F_{RD_i} = -I_i^{-1} S_{\omega_i} I_i \begin{Bmatrix} p_i \\ q_i \\ r_i \end{Bmatrix} + I_i^{-1} \begin{Bmatrix} M_{X_i} \\ M_{Y_i} \\ M_{Z_i} \end{Bmatrix} \quad (14)$$

and

$$S_{\omega_i} = \begin{bmatrix} 0 & -r_i & q_i \\ r_i & 0 & -p_i \\ -q_i & p_i & 0 \end{bmatrix} \quad (15)$$

Equations (9)–(14) are the well known rigid six DOF equations of motion. Note that I_i is the mass moment of inertia of the i th rigid body about its mass center. The total externally applied forces and moments about the mass center in the i th body reference frame are

given as $[F_{X_i} \ F_{Y_i} \ F_{Z_i}]^T$ and $[M_{X_i} \ M_{Y_i} \ M_{Z_i}]^T$, respectively. These externally applied loads do not include the effect of connection constraints. They are, however, a function of the system state and control input.

C. Effects of Joint Connections on Dynamic Equations

In Eq. (4), the term $G_i U$ transforms the single rigid body formulation to a multibody dynamics formulation. Notice that U is a vector of connection constraint forces and moments from all the connection elements of the system, not just connections associated with the i th rigid body. Each joint is viewed as connecting a parent and child body together. For the block column in G associated with joint j , the following two entries exist for the parent and child. For the parent body, $G_i = G_{P_j}$. For the child body, $G_i = G_{C_j}$. For all other bodies, $G_i = 0$.

$$G_{P_j} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m_{P_j}} T_{P_j}^T \Gamma_{T_j} & 0 \\ I_{P_j}^{-1} S_{P_j} T_{P_j}^T \Gamma_{T_j} & I_{P_j}^{-1} T_{P_j}^T \Gamma_{R_j} \end{bmatrix} \quad (16)$$

$$G_{C_j} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\frac{1}{m_{C_j}} T_{C_j} T_{P_j}^T \Gamma_{T_j} & 0 \\ -I_{C_j}^{-1} S_{C_j} T_{C_j} T_{P_j}^T \Gamma_{T_j} & -I_{C_j}^{-1} T_{C_j} T_{P_j}^T \Gamma_{R_j} \end{bmatrix} \quad (17)$$

The matrices G_{P_j} and G_{C_j} populate the block column of the G matrix associated with the j th joint. The matrix is then populated for all M joints. Note that m_{P_j} , m_{C_j} , I_{P_j} , and I_{C_j} are the mass of the parent body, mass of the child body, inertia matrix of the parent body, and inertia matrix of the child body, respectively, for the j th joint. Also, S_{P_j} is a skew symmetric cross product operator matrix acting on the distance vector from the mass center of the parent body to the joint location point on the parent body.

$$S_{P_j} = \begin{bmatrix} 0 & -\Delta Z_{P_j} & \Delta Y_{P_j} \\ \Delta Z_{P_j} & 0 & -\Delta X_{P_j} \\ -\Delta Y_{P_j} & \Delta X_{P_j} & 0 \end{bmatrix} \quad (18)$$

The matrix S_{C_j} is similar to S_{P_j} but for the child body. The matrices Γ_{T_j} and Γ_{R_j} ensure the correct scalar force, and moment loads for a given joint are applied to the body through the correct body frame vector components. Their selection will be detailed in the next section.

To form the overall system equations, the differential equations represented by Eq. (4) are concatenated for all bodies to form a set of nonlinear ordinary differential equations that are affine in the force and moment constraint vector, U .

$$\dot{X} = F + GU \quad (19)$$

Here, F contains the individual rigid body equations of motion and is the kernel on which the method is based. The term GU glues the multibody system together through joint connection constraint forces and moments. The exact form of this term varies depending on the joint and is discussed in detail below. This set of differential equations is simultaneously integrated forward in time, allowing for the simulation of multiple bodies concurrently.

D. Generic Joint Description and Resulting Geometric Constraints

The joints, as depicted in Fig. 1, are introduced between two bodies to join the bodies together in a physically representative manner. Each joint connection stipulates constraints on the connected bodies that must be satisfied. There are two cases of motion constraints imposed by joint connections, namely, translational and rotational. Each joint constraint reduces the number of degrees of freedom in the system by an increment of one. Table 1 gives five examples of joints. For instance, a simple hinge is a revolute joint and can only rotate in one direction. Thus, it is constrained in all three translational directions and is constrained along the other two rotational axes, resulting in the three translation constraints and two rotational constraints in the table. Likewise, a gimbal joint, which is a spherical joint, permits rotation about three axes and has zero rotational constraints. A gimbal joint is not allowed to translate along any direction and thereby has three translation constraints. See Bauchau and Laulusa [23] for more details.

For each translational constraint that exists, an element must be added to the Γ_{T_j} matrix. Likewise, for each rotational constraint that exists, an element must be added to the Γ_{R_j} matrix. The translational and rotational Γ matrices serve to apply the correct forces and moments to each body depending on the type of joint and which direction it is aligned with respect to the parent and child bodies. In general, a Γ matrix will always have three rows defining the direction of action and a number of columns equal to the number of constraints. The exception is when there are no constraints and the Γ matrix is empty.

Constraint translational errors are calculated by first finding the difference in position between the joint location on the parent body and the joint location on the child body. This vector is then dotted with an appropriate vector from the parent joint reference frame or child joint reference frame. The resulting scalar is a constraint error. This is repeated for each translational degree of freedom restricted by the joint. For rotational joints, appropriate unit vectors from the parent joint reference frame and child joint reference frame are dotted together to form a scalar constraint error.

1. Translational

For a translational joint constraint, a fundamental quantity of interest is the position vector from the joint point on the parent body to the joint point on the child body. One or more of the measure numbers, also known as components, of this vector must be held to zero to satisfy a translational joint motion constraint. Mathematically, for joint j , this can be written as follows.

$$\mathbf{e}_{T_j} = \mathbf{r}_{O \rightarrow PJ} - \mathbf{r}_{O \rightarrow CJ} \quad (20)$$

This vector is expressed in the parent joint reference frame, and the matrix $\Gamma_{T_j}^T$ is applied to it to isolate only those measure numbers to be

constrained. The measure numbers in the parent joint frame in more detail are given by Eqs. (21) and (22).

$$E_{T_j} = T_{PJ} T_{P_j} \begin{Bmatrix} x_{P_j} - x_{C_j} \\ y_{P_j} - y_{C_j} \\ z_{P_j} - z_{C_j} \end{Bmatrix} + T_{PJ} \begin{Bmatrix} \Delta X_{P_j} \\ \Delta Y_{P_j} \\ \Delta Z_{P_j} \end{Bmatrix} - T_{PJ} T_{P_j} T_{C_j}^T \begin{Bmatrix} \Delta X_{C_j} \\ \Delta Y_{C_j} \\ \Delta Z_{C_j} \end{Bmatrix} \quad (21)$$

$$E_{T_j}^* = \Gamma_{T_j}^T E_{T_j} \quad (22)$$

2. Rotational

For a rotational joint constraint, a fundamental quantity of interest is the relative rotation of the child joint frame with respect to the parent joint frame. One or more dot products between vectors in the parent joint and child joint frames must be held to zero to satisfy a rotational joint motion constraint. By expressing both unit vectors in the inertial reference frame, any dot product condition can be written as shown in Eq. (24).

$$E_{R_j} = T_{PJ} T_{P_j} T_{C_j}^T T_{C_j}^T \quad (23)$$

$$E_{R_j}^* = \Phi_j T_{PJ} T_{P_j} T_{C_j}^T T_{C_j}^T \Psi_j^T = \Phi_j E_{R_j} \Psi_j^T \quad (24)$$

In the above equation, Φ_j and Ψ_j are used to select the correct error components from the matrix of dot product conditions [23].

In the constraint stabilization algorithm described in Sec. II.E, derivatives of the constraint vector error are required. These derivatives are presented in the Appendix. To aid in bookkeeping, the translational and rotational errors from all joints of the system are placed into a large system constraints error vector, $E(X)$.

$$E(X) = [E_{T_1}^T \quad E_{R_1}^T \quad \cdots \quad E_{T_M}^T \quad E_{R_M}^T]^T \quad (25)$$

It should be noted that the analyst must ensure that the specified joint is not redundant. This is true for both translation and rotation joint constraints.

E. Constraint Stabilization

When collected together, Eqs. (19) and (25) represent a set of differential algebraic equations. The vector U contains all scalar constraint force and moment components from all joints in the system, and the vector E is a vector of constraint equations that must be nulled at all times. The number of constraint equations is exactly equal to the number of constraint force and moment scalars. In control theory language, if U is viewed as the control vector and E is viewed as the output of the system to be controlled, the system is noted as square because the number of inputs and outputs is equal. Recall that the purpose of the constraint force and moment inputs is to zero the constraint equations. Thus, U can be viewed as a control where we seek to solve for U so that it nulls Eq. (25). If the control system is designed to be stable and begins by satisfying the constraint equation $E(X(t = t_0)) = 0$, the equations of motion can be integrated with an ordinary differential equation solver while satisfying the constraint equations. This constraint stabilization controller is termed the “glue code controller” because it determines the constraint loads that properly connect, or “glue,” the system together.

A nonlinear controller is defined by first considering the dynamics of the constraint equations and taking derivatives of the outputs until controls, U , appear.

$$\dot{E}(X) = \frac{\partial E}{\partial X} \frac{\partial X}{\partial t} = \frac{\partial E}{\partial X} (F + GU) = \frac{\partial E}{\partial X} F + \frac{\partial E}{\partial X} GU \quad (26)$$

For holonomic constraints, the last term in Eq. (26) will be zero, because the constraint error only depends on the position and

Table 1 Example joints number of constraints

Joint type	Number of translational constraints	Number of rotational constraints
Revolute	3	2
Prismatic	2	3
Cylindrical	2	2
Planar	1	2
Spherical	3	0

orientation of each body. Then the partial derivatives with respect to velocity and angular rate components are all zero. The nonzero blocks in the constraint derivative matrix are multiplied by the zero blocks of the G matrix, whereas the zero blocks of the constraint derivative matrix multiply the nonzero blocks of the G matrix. This is because GU affects the kinetic dynamic equations, whereas the constraint errors only depend on the kinematics. Taking a second derivative of the constraint equations yields:

$$\ddot{E}(X) = \frac{\partial \dot{E}}{\partial X} \frac{\partial X}{\partial t} = \frac{\partial \dot{E}}{\partial X} (F + GU) = \frac{\partial \dot{E}}{\partial X} F + \frac{\partial \dot{E}}{\partial X} GU \quad (27)$$

Defining

$$\tilde{F} = \frac{\partial \dot{E}}{\partial X} F \quad (28)$$

$$\tilde{G} = \frac{\partial \dot{E}}{\partial X} G \quad (29)$$

yields the following second order dynamics for the constraint equations.

$$\ddot{E}(X) = \tilde{F}(X) + \tilde{G}(X)U(X) \quad (30)$$

Because the matrix \tilde{G} is almost always nonsingular, the relative degree for each output of the system is two. The error dynamics given in Eq. (30) are affine in the constraint force and moment vector U , and the system is square such that the number of controls is equal to the number of quantities to control. This form permits direct computation of the constraint loads provided \tilde{G} is not singular. Numerous nonlinear control law design techniques are suitable to solve this problem, two of which are feedback linearization and sliding mode control [28].

1. Feedback Linearization Formulation

Setting the right hand side of Eq. (30) equal to the psuedo control, γ :

$$\ddot{E}(X) = \gamma \quad (31)$$

$$\gamma = \tilde{F} + \tilde{G}U \quad (32)$$

The pseudo control is selected so that the constraint equation dynamics are exponentially stable. If the constraint equations have an error initially or if numerical round off error causes the constraint equations to become slightly violated, then the constraint equations will tend back to zero. The pseudo control is selected in the following manner.

$$\gamma = -2\xi\omega_n\dot{E} - \omega_n^2 E \quad (33)$$

Then the scalar constraint force and moment vector, U , is given below.

$$U = -\tilde{G}^{-1}(2\xi\omega_n\dot{E} + \omega_n^2 E + \tilde{F}) \quad (34)$$

With the constraint force and moment vector defined above, the constraint error dynamic equations become an uncoupled set of simple damped oscillators.

$$\ddot{E} + 2\xi\omega_n\dot{E} + \omega_n^2 E = 0 \quad (35)$$

By picking the damping ratio and the natural frequency properly, Eq. (35) will be stable. This is a standard feedback linearization controller where the zero dynamics represent the dynamics of the properly coupled physical system.

2. Sliding Mode Control Formulation

A sliding surface is defined in terms of the constraint error vector and is defined to be the same size as the constraint error vector.

$$S = \dot{E} + \Lambda E \quad (36)$$

The matrix Λ is diagonal and stable where the i th diagonal entry is λ_i . Given an initial nonzero value of S , a constraint stabilization controller is defined that drives S toward zero in finite time using a relay function with a boundary layer.

$$U = -\tilde{G}^{-1} \left(\tilde{F} + \Lambda \dot{E} + K \text{Sat} \left(\frac{S}{\delta} \right) \right) \quad (37)$$

Once the system has achieved sufficiently small values of the sliding surface, the constraint error dynamics are given by a simple stable homogeneous first order system that tends to decay the constraint error to zero. The function $\text{Sat}(\cdot)$ is the saturation function.

$$\text{Sat} \left(\frac{S}{\delta} \right) = \begin{cases} S/\delta & |S/\delta| \leq 1 \\ \delta & |S/\delta| > 1 \end{cases} \quad (38)$$

Collected together, Eqs. (37) and (38) represent a standard sliding mode controller with a boundary layer to eliminate high frequency chattering.

F. Computational Considerations

To calculate the constraint forces and moments U for either formulation above, the derivatives for the $\frac{\partial \dot{E}}{\partial X}$ matrix are necessary. The analytical derivatives are given in the Appendix of this paper. These partial derivatives can be quite involved and are prone to coding mistakes. As an alternative approach, partial derivatives were calculated numerically with a fourth order center difference formula.

$$\begin{aligned} \frac{\partial \dot{E}_i}{\partial X_j} \Big|_{\bar{X}_j} &\approx \frac{\dot{E}_i(\bar{X}_j - 2\delta X_j) - 8\dot{E}_i(\bar{X}_j - \delta X_j) + 8\dot{E}_i(\bar{X}_j + \delta X_j) - \dot{E}_i(\bar{X}_j + 2\delta X_j)}{12\delta X_j} \end{aligned} \quad (39)$$

Numerical derivatives are much easier to code but cause an increase in run time because of the many calls to the \dot{E} function. For example, for a simulation with three bodies with 13 states each, the \dot{E} function must be called 40 times (39 for varying each of the states and once for a baseline).

The calculation of the glue code is the component of the simulation code that hampers run time the most. To decrease run time, for either analytical or numerical derivatives, the calculation of the entire $\frac{\partial \dot{E}}{\partial X}$ matrix can be skipped so that it does not update every time step. This will clearly decrease the accuracy of the simulation. However, when updated every time step, the constraint errors are often kept to an unnecessarily small value, such as 10^{-18} , whereas the simulation will be sufficiently accurate when the constraint errors are kept to a larger, but still small, value such as 10^{-6} . The calculation of the constraint forces and moments can be skipped the maximum number of time steps that will still keep the errors below an allowable threshold. There is a trade-off between accuracy and run time that, when balanced, can yield a powerful yet fast simulation tool.

G. Elastic Joints

Elastic joints are modeled in a similar manner to joint constraints. The constraint forces are calculated through a constraint stabilization scheme to keep the joint error terms very small, whereas elastic joint forces are simply calculated to be proportional and in an opposite direction to the joint error terms (linear spring and damper).

$$U_E = k(E - E_0) + c\dot{E} \quad (40)$$

1. Translational

The translational error terms used for constraint stabilization are in fact the relative translation between the joints, so these terms can be

used directly to calculate the elastic joint force. For an elastic translation connection along the direction μ , the scalar elastic force is given below, where k_T is the spring constant, c_T is the damping constant, and E_{T0} is the zero load elastic deflection.

$$F_E = k_T(E_T(\mu) - E_{T0}) + c_T \dot{E}_T(\mu) \quad (41)$$

The effect of the translational elastic forces and moments on the parent and child body dynamic equations is similar to the joint constraints described earlier. Let the vector U_{TE} denote the total translational elastic force for a particular joint. Because each joint connects a parent and child body together, the elastic joint forces act on both the parent and child bodies. The equations below detail the force contribution to F_{FD_i} and the moment contribution to F_{RD_i} for a parent body.

$$C_{P_j}(\mathbf{F}_E) = T_{PJ}^T \Lambda_{T_j} U_{TE} \quad (42)$$

$$C_{P_j}(\mathbf{M}_E) = S_{P_j} T_{PJ}^T \Lambda_{T_j} U_{TE} \quad (43)$$

The matrix Λ_{T_j} converts the scalar elastic forces to vector measure numbers along the joint axis. The corresponding equations for the child body are given below.

$$C_{C_j}(\mathbf{F}_E) = -T_{C_j} T_{P_j}^T T_{PJ}^T \Lambda_{T_j} U_{TE} \quad (44)$$

$$C_{C_j}(\mathbf{M}_E) = -S_{C_j} T_{C_j} T_{P_j}^T T_{PJ}^T \Lambda_{T_j} U_{TE} \quad (45)$$

2. Rotational

The rotational error terms used for constraint stabilization are taken from matrices of the dot products of the basis vectors of the parent and child joint reference frames and its time derivative, E_R^T and \dot{E}_R^T . The angle separating the parent and child joint frames along the elastic axis is extracted from this matrix of dot products. The angular separation between the parent and child joint along a particular joint axis is determined as follows.

$$\varepsilon = \tan^{-1} \left(\frac{E_R^T(\mu, \eta) - E_R^T(\eta, \mu)}{E_R^T(\mu, \mu) + E_R^T(\eta, \eta)} \right) \quad (46)$$

For an **I** axis rotation, $\mu = 2, \eta = 3$. For a **J** axis rotation, $\mu = 3, \eta = 1$. For a **K** axis rotation, $\mu = 1, \eta = 2$. Once the angular separation and the rate of change of this angle have been determined, the scalar elastic restoring moment can be calculated assuming a linear rotational spring and damper:

$$M_E = -k_R(\varepsilon - \varepsilon_{R0}) - c_R \dot{\varepsilon} \quad (48)$$

The effect of the rotational elastic moments on the parent and child body dynamic equations is similar to the joint constraints described earlier. Let the vector U_{RE} denote the total rotational elastic moment for a particular joint. Because each joint connects a parent and child body together, the elastic joint forces act on both the parent and child bodies. The equations below detail the moment contribution to F_{RD_i} for a parent body.

$$C_{P_j}(\mathbf{M}_E) = T_{PJ}^T \Lambda_{R_j} U_{RE} \quad (49)$$

The corresponding equations for the child body are given below.

$$C_{C_j}(\mathbf{M}_E) = -T_{C_j} T_{P_j}^T T_{PJ}^T \Lambda_{R_j} U_{RE} \quad (50)$$

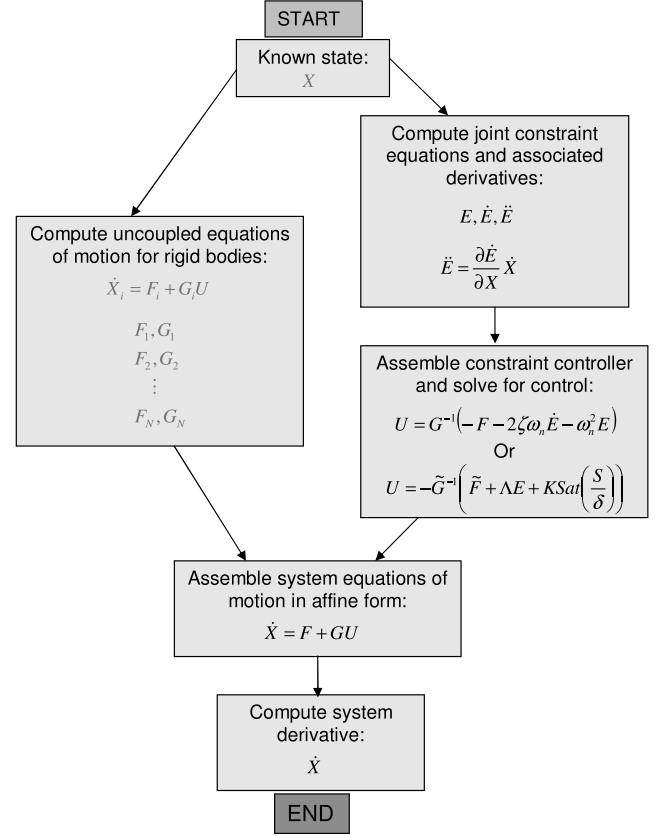


Fig. 2 Block diagram detailing the flow of the method in a simulation.

H. Simulation

Figure 2 depicts the process of using the above methodology and equations in a simulation for one time step. Given a known state, an initial condition or the previous state, the uncoupled six DOF equations of motion are calculated for each body. Additionally, the G matrices for each body are calculated. This is the block of the left branch. The right branch depicts the glue code, which calculates the

$$\dot{\varepsilon} = \frac{(E_R^T(\eta, \mu) - E_R^T(\mu, \eta))(\dot{E}_R^T(\mu, \mu) + \dot{E}_R^T(\eta, \eta)) + (E_R^T(\eta, \eta) + E_R^T(\mu, \mu))(\dot{E}_R^T(\mu, \eta) - \dot{E}_R^T(\eta, \mu))}{4} \quad (47)$$

joint constraint errors and error derivatives and uses them to determine the constraint forces and moments, U . The results of the right branch and left branch are then combined to determine the system derivatives for each body, and these results are concatenated into a single vector. At the end, the set of system derivatives \dot{X} can then be integrated forward in time. This process is then repeated until the end of the simulation time is reached.

I. Linear Model Analysis

It is often useful to construct a linear dynamic model of a system to perform modal analysis or frequency domain computation. Closed form solutions of the linear system given initial conditions are always possible. Additionally, global stability can be determined from the eigenvalues of the state matrix, generally denoted as A . The process for generating the linear system of an unconstrained dynamic system is fairly straight forward as the Jacobian of the state equations is computed. In formulations using constrained coordinates, such as this multibody method, the process to generate the linear system is more involved, because the fully dimensioned system must be

projected down to the lower dimension unconstrained system. To construct a linear model, the constrained equations of motion and the constraint equations are linearized, and by cleverly substituting part of the linearized constraint equations into part of the linearized differential equations, we can form a linear model of the unconstrained system. This process is detailed below, starting with linearized constraint equations of motion and constraint equations in Eqs. (51) and (52).

$$\delta\dot{X} = A\delta X + B\delta U_C \quad (51)$$

$$C\delta X + D\delta U_C + E = 0 \quad (52)$$

The state vector is split into two parts with sizes associated with a set of constrained coordinates. The vector Z_U consists of the unconstrained states and the vector Z_C consists of the constrained states. A transformation matrix, T , is formed to transform the original state vector X into the split state vector consisting of Z_U and Z_C . By definition, T is invertible.

$$\begin{Bmatrix} Z_U \\ Z_C \end{Bmatrix} = [T]\{X\} \quad (53)$$

$$\{X\} = [T]^{-1} \begin{Bmatrix} Z_U \\ Z_C \end{Bmatrix} \quad (54)$$

Change coordinates to state variables $Z = [Z_U, Z_C]$ to obtain the equations below.

$$T^{-1}\delta\dot{Z} = AT^{-1}\delta Z + B\delta U_C \quad (55)$$

$$CT^{-1}\delta Z + D\delta U_C + E = 0 \quad (56)$$

or

$$\delta\dot{Z} = (TAT^{-1})\delta Z + (TB)\delta U_C \quad (57)$$

$$\delta\dot{Z} = (TAT^{-1})\delta Z + (TB)\delta U_C \quad (58)$$

$$C_Z\delta Z + D_Z\delta U_C + E_Z = 0 \quad (59)$$

Expanding:

$$\begin{Bmatrix} \delta\dot{Z}_U \\ \delta\dot{Z}_C \end{Bmatrix} = \begin{bmatrix} A_{Z11} & A_{Z12} \\ A_{Z21} & A_{Z22} \end{bmatrix} \begin{Bmatrix} \delta Z_U \\ \delta Z_C \end{Bmatrix} + \begin{bmatrix} B_{Z1} \\ B_{Z2} \end{bmatrix} \delta U_C \quad (60)$$

$$[C_{Z1} \quad C_{Z2}] \begin{Bmatrix} \delta Z_U \\ \delta Z_C \end{Bmatrix} + D_Z\delta U_C + E_Z = 0 \quad (61)$$

Substitution of the linearized constraint equations, rewritten as Eq. (62), into the equations of motion of the unconstrained states, Eq. (63), eliminates the extra state equations (Z_C) and yields the linearized unconstrained equations of motion, Eq. (64).

$$\delta Z_C = -C_{Z2}^{-1}(C_{Z1}\delta Z_U + D_Z\delta U_C + E_Z) \quad (62)$$

$$\delta\dot{Z}_U = A_{Z11}\delta Z_U + A_{Z12}\delta Z_C + B_{Z1}\delta U_C \quad (63)$$

$$\begin{aligned} \delta\dot{Z}_U &= (A_{Z11} - A_{Z12}C_{Z2}^{-1}C_{Z1})\delta Z_U \\ &+ (B_{Z1} - A_{Z12}C_{Z2}^{-1}D_Z)\delta U_C - A_{Z12}C_{Z2}^{-1}E_Z \end{aligned} \quad (64)$$

The term $(A_{Z11} - A_{Z12}C_{Z2}^{-1}C_{Z1})$ is the state matrix of the unconstrained system, which can then be evaluated for the stability of the system.

III. Example Applications

To demonstrate how the above methodology can be used in simulation of multibody air vehicle configurations, three example applications are presented below, a projectile with an internal translating mass, an airdrop system using a parafoil canopy, and an articulated wing aircraft.

A. Projectile Equipped with an Internal Translating Mass

An effective control mechanism for some smart projectile applications is translation of a relatively small mass inside a cavity within the projectile. By oscillating the mass at the projectile spin frequency, appreciable and controllable changes in the trajectory can be realized. To predict these trajectory changes along with actuator power requirements dictates that the flight dynamic model contains degrees of freedom associated with projectile six DOF motion as well as a degree of freedom associated with the motion of the internal mass inside the projectile. Figure 3a shows an overall schematic of the system, whereas Fig. 3b presents a detailed view of the local area near the internal translating mass (ITM).

This system has one slider joint with the parent body defined as the translating mass and the child body defined as the projectile. Notice that the internal mass moves relative to the projectile body axis along \mathbf{J}_{P_j} . Because all the body based reference frames are aligned, $T_{P_j} = T_{C_j}$. The joint transformation matrices are identity matrices, $T_{P_j} = T_{C_j} = I$. Because movement of the internal translating mass is along $\mathbf{J}_{P_j} = \mathbf{J}_{C_j}$, two scalar nonzero constraint force components exist at the joint along $\mathbf{I}_{P_j} = \mathbf{I}_{C_j}$ and $\mathbf{K}_{P_j} = \mathbf{K}_{C_j}$. Because the projectile body and the internal translating mass reference frames must stay aligned because only translational motion is permitted between the two bodies, three scalar nonzero constraint moment components exist along all three directions.

$$\mathbf{F}_C = F_{XC}\mathbf{I}_{P_j} + F_{ZC}\mathbf{K}_{P_j} \quad (65)$$

$$\mathbf{M}_C = M_{XC}\mathbf{I}_{P_j} + M_{YC}\mathbf{J}_{P_j} + M_{ZC}\mathbf{K}_{P_j} \quad (66)$$

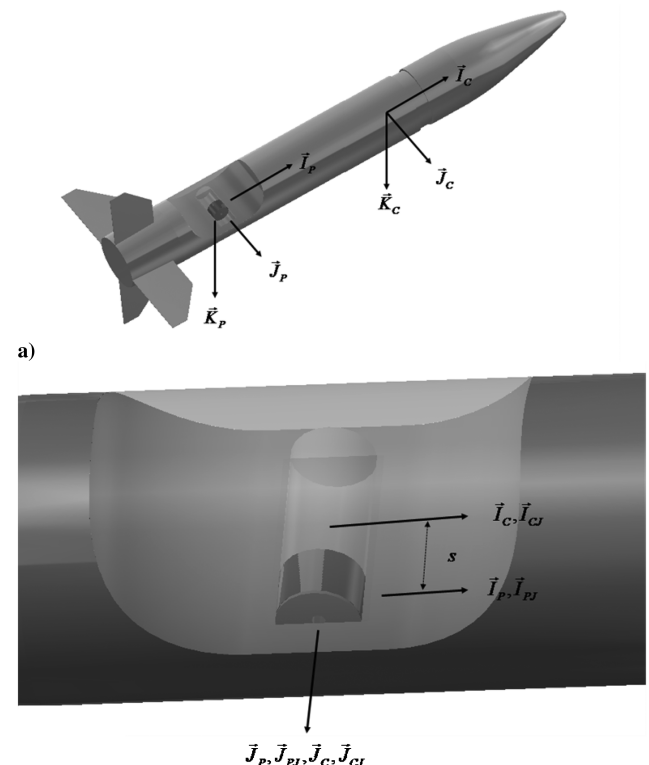


Fig. 3 Internal translating mass: a) overall schematic and b) detailed schematic.

Table 2 Properties of projectile

Reference diameter	0.1 m
Mass	17.61 kg
I_{xx}	$0.0377 \text{ kg} \cdot \text{m}^2$
I_{yy}	$0.8533 \text{ kg} \cdot \text{m}^2$
I_{zz}	$0.8533 \text{ kg} \cdot \text{m}^2$
Mass center distance from base of round	0.0305 m

Table 3 Properties of internal translating mass

Mass	0.73 kg
I_{xx}	$0.00000184 \text{ kg} \cdot \text{m}^2$
I_{yy}	$0.00000340 \text{ kg} \cdot \text{m}^2$
I_{zz}	$0.00000184 \text{ kg} \cdot \text{m}^2$

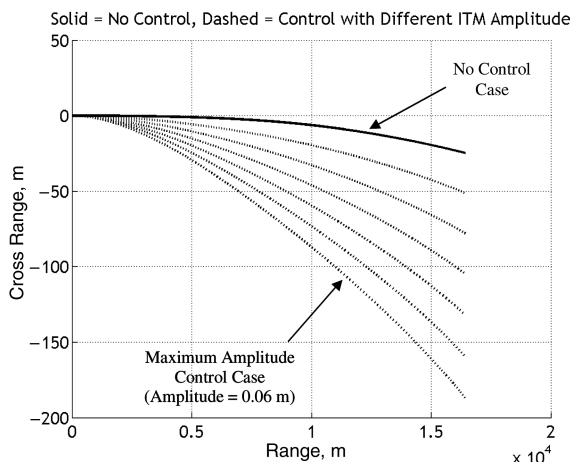
This leads to

$$\Gamma_{T_j} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad \Gamma_{R_j} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (67)$$

A total of five geometric constraint equations must be satisfied for this configuration, namely, two translational constraint equations and three rotational constraint equations.

The multibody flight dynamic modeling scheme discussed above was used to simulate this system for a fin stabilized projectile. Basic properties of the projectile and internal translating mass are given in Tables 2 and 3. The ITM cavity is located at the main projectile mass center stationline.

For the example results shown below, the projectile is launched with an Euler roll angle of 180 deg, an Euler pitch angle of 44.7 deg, and an Euler yaw angle of 0.0 deg. The initial velocity of the projectile is 860 m/s, whereas the initial spin rate is 5 rad/s. A total of seven trajectories are presented. The first case, shown as a solid black line, represents a rigid projectile where the ITM is held fixed in the center of the cavity (no control), whereas all other cases (shown as dashed lines) represent a situation where the internal mass is vibrated in the cavity at the roll rate of the projectile to achieve trajectory changes (control). Figure 4 presents trajectory results and indicates typical projectile behavior for an indirect fire projectile. The projectile achieves a range of 16 km and has a total velocity of 300 m/s at impact. Substantial cross range changes are noted, on the order of 150 m over the trajectory depending on the amplitude of internal mass vibration (Fig. 5). Figure 6 shows the norm of the translation and rotation constraint errors. Given that the translational constraint error is bounded to 1.5×10^{-10} and the rotational

**Fig. 4** Cross range vs range for ITM projectile.

constraint error is bounded by 5×10^{-15} , the constraint stabilization control works well.

In this example application, using numerical derivatives and skipping glue code calculations, as described in Sec. II.F, is investigated. As a baseline, the projectile is first simulated with no joint connections. Then the run time and maximum error norms of the translational and rotational joints are compared for both analytical and numerical derivatives, as well as for skipping updates of the constraint error calculations by zero, one, two, three, and four time steps for both analytical and numerical derivatives. The results are

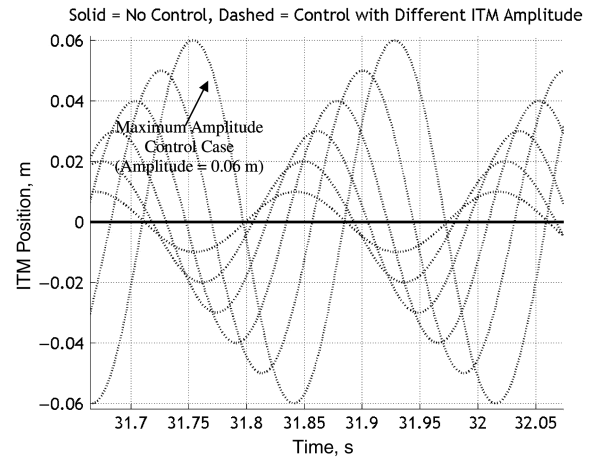
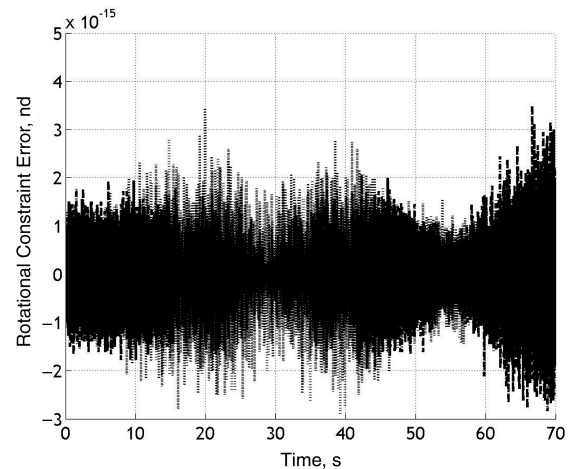
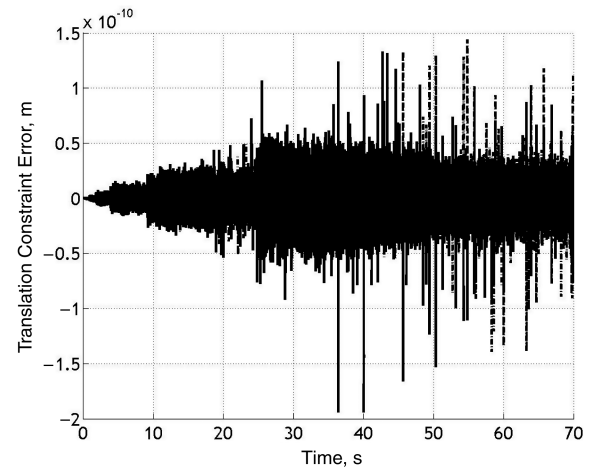
**Fig. 5** ITM displacement vs time for ITM projectile (short time window).**Fig. 6** Translation and rotation constraint error norm for ITM projectile.

Table 4 Runtime for various simulation methods

ISKIP	ANALYTIC			NUMERIC		
	User CPU time	Translational error norm	Rotational error norm	User CPU time	Translational error norm	Rotational error norm
0 (No connection)	14.02			14.02		
0	202.96	1.12e − 10	2.05e − 15	935.47	1.12e − 10	1.88e − 15
1	119.27	6.98e − 5	9.98e − 8	501.26	6.98e − 5	9.98e − 8
2	85.11	2.58e − 4	2.69e − 7	343.4	2.58e − 4	2.69e − 7
3	69.6	5.44e − 4	4.62e − 7	256.9	5.44e − 4	4.62e − 7
4		Unstable			Unstable	

shown in Table 4, where the variable ISKIP is the number of time steps the constraint error calculations are skipped. The relative run times for the various methods coded in FORTRAN on an Intel-based Windows machine are shown below. In these simulations, the projectile was simulated with the amplitude of the internal translating mass at 0.03 m. The fourth order Runge–Kutta integrator ran with a time step of 0.00005 s. Translational and rotational joint errors are evaluated separately because of their different units and orders of magnitude.

When joint connections are turned off, the user CPU time is 14.02 s, which is drastically lower than any user CPU time shown in the table with the joints connected. Thus, the earlier assertion that the glue code slows the run time is verified. Run time can be reduced by skipping updates of the constraint forces and moments. The skipping can also be selected so that the error norms stay below an acceptable threshold. The skipping rate also depends on the time step of the integrator. For example, if the time step is extremely small, the calculation of the glue code can likely be skipped more than for larger time steps. As shown above, the analytic derivatives for $\frac{\partial \mathbf{E}}{\partial \mathbf{X}}$ run faster than their numeric counterpart for the same amount of error. However, the numeric derivatives are easier to code, leading to faster development time and a simulation less prone to error.

B. Airdrop with Parafoil Canopy

Parafoil canopies are a type of gliding parachute that make use of ram-air to inflate a wing-like structure made of fabric. A payload is attached to the canopy through a network of rigging lines. To minimize the effect of asymmetries in the internal payload weight distribution, the payload is often attached to the canopy rigging via a single confluence point, as shown in Fig. 7. The rigging attachment allows free rotation of the payload and canopy in all axes about this confluence point. The system is modeled as two rigid bodies, the canopy and payload, connected by a gimbal joint at the rigging confluence point. With the exception of the rotation about this joint,

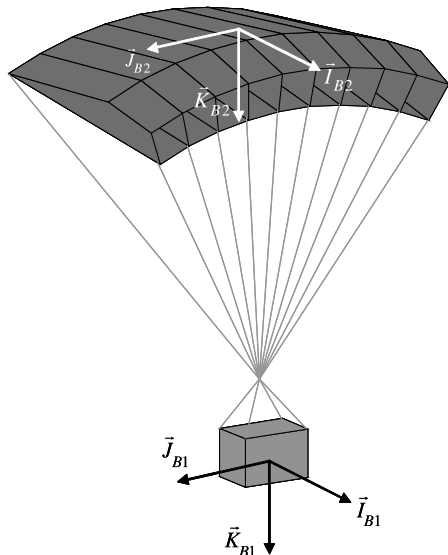


Fig. 7 Example of parafoil and payload aircraft.

the rigging geometry is assumed to be rigid. This setup results in translational forces along all three axes, enforcing the confluence point and no constraint moments, leading to

$$\Gamma_{T_j} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \Gamma_{R_j} = [] \quad (68)$$

For this case, simulation results are generated using the multibody method described in the current work and compared with results generated with an unconstrained coordinate approach. A complementary simulation model using a minimal coordinate approach is given in Ref. [8]. The approach is based on a Newton–Euler derivation of the equations of motion to obtain a nine DOF parafoil and payload simulation model. Both simulations were performed using fourth order Runge–Kutta integration with a fixed time step of 0.02 s. The payload mass is 264 kg, and the parafoil surface area is 54 m². Mass, geometry, and aerodynamic data for the parafoil and payload system were set to match flight test data from an experimental airdrop program. The simulation demonstrates a step input to the right trailing edge of the canopy, resulting in a turn to the right. The system is initially in straight, gliding flight. At $t = 5$ s, the right trailing edge of the canopy is deflected 1 m down, causing the system to enter the a right turn. At $t = 15$ s, the control input is released, and the canopy returns to straight gliding flight.

Figure 8 shows the roll angle and roll rate of both the payload and canopy using the two simulation methods. The results obtained using the minimal coordinate approach are designated “9 dof” in the figures, and the results obtained with the constrained coordinate

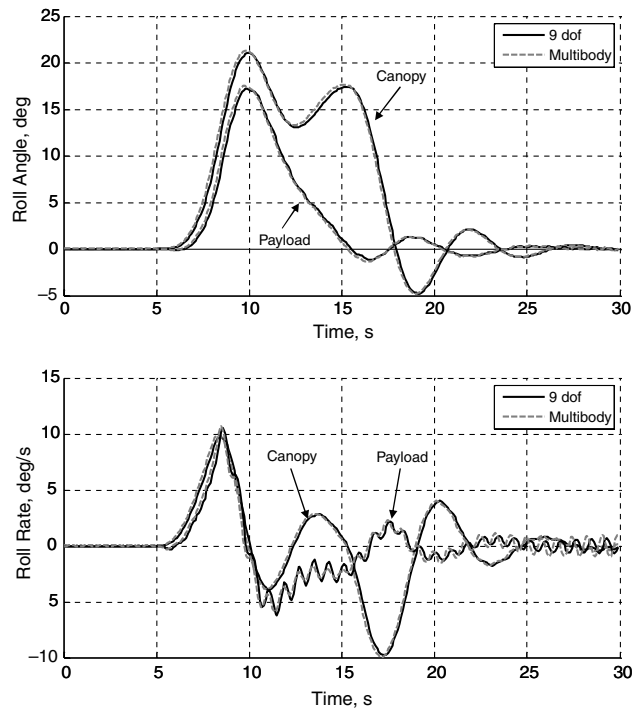


Fig. 8 Roll angle and roll rate vs time for parafoil and payload simulation.

approach described in the current work are designated “multibody.” The results show that there is a significant amount of relative motion between the canopy and payload during the turn. The results are essentially identical for the unconstrained, nine DOF model and the constrained coordinate approach described in the present work. This is expected because the constraint controller keeps any errors in the constraint equations extremely small, resulting in an excellent approximation of the nine DOF model. Of course, the constrained coordinate approach is more computationally intensive, because the constraint forces and moments must be calculated analytically at each time step to ensure that the constraints are satisfied, whereas the constraints are satisfied by definition in the unconstrained coordinate approach. For the example simulation, the nine DOF, unconstrained coordinate approach took an average of 0.59 s, whereas the constrained coordinate approach described by the current work took an average of 1.87 s to run. This indicates that the constrained coordinate approach described here results in a factor of 3 increase in computation time compared with a minimal coordinate approach for a two body system. However, the method presented here has the advantage of using the standard six DOF equations of motion as the kernel of the method, which greatly reduces setup time for complex simulations involving many joints and bodies. This also makes it appealing to a much broader range of engineers who may not have formal training in multibody dynamic formulations. In addition, it is relatively straight forward to create a generic N body code with any number and type of joint connections using this method, which is not the case for any minimal coordinate methods.

The multibody parafoil and payload simulation was originally run using the feedback linearization controller for constraint stabilization. The natural frequency and damping ratio of the feedback linearization controller were set to 5 rad/s and 1.0, respectively. For comparison, the simulation was run a second time using a sliding mode controller with $\lambda = 4 \text{ s}^{-1}$, $k = 12 \text{ m/s}^2$, and $\delta = 10^{-10} \text{ m/s}$. The translational error norms for the canopy-to-payload joint are compared in Fig. 9. To create a challenging scenario for the constraint controller, the initial position of the canopy and payload are set such that the initial translational joint error is 1 m. The performance of the feedback linearization and sliding mode controllers is nearly identical. Despite the simulation starting with a large initial joint error of 1 meter, both controllers quickly stabilize the joints to fractions of a millimeter. The slight increase in joint error beginning at 5 s is due to the control input. As the simulation progresses, the joint error for the sliding mode controller levels off at a slightly higher value because the constraint error approaches the level established by the sliding mode boundary layer. However, any differences in joint error are so small that there is no discernible difference in the simulated state variables. This simulation comparison demonstrates that essentially identical performance can be obtained by tuning a suitable nonlinear control technique to perform the constraint stabilization.

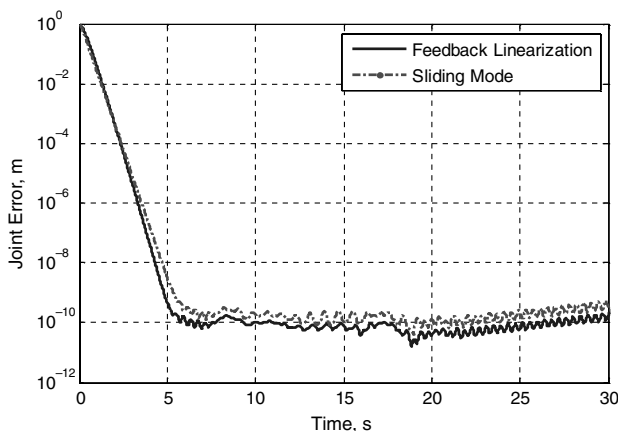


Fig. 9 Constraint error vs. time.

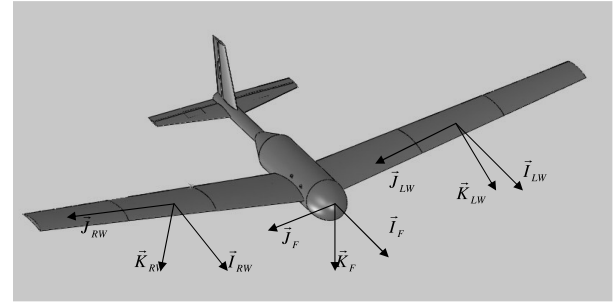


Fig. 10 Schematic of the articulated aircraft.

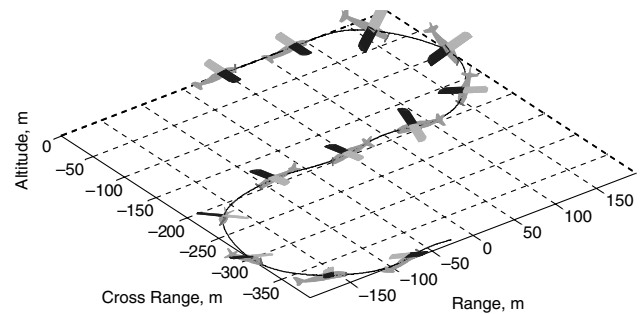
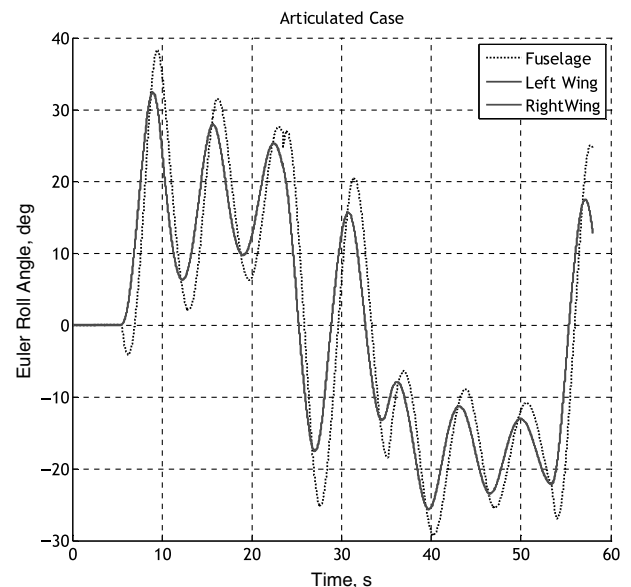


Fig. 11 Commanded way points and trajectory for maneuver.

C. Articulated Micro Air Vehicle

The above multibody flight simulation technique can also be used to simulate micro air vehicles with articulated structures. In this example, the left and right wing of the vehicle are hinged at the root chord to allow them to flap up and down, changing the dihedral angle. The wings can be passive or active structures. The flapping response is controlled by a spring and damper at the hinge, and there are also stops to prevent the wing from flapping past a certain angle. Small air vehicles are sensitive to gusts, and this articulation can alleviate the aircraft's response to gusts, which is important for camera pointing, tracking, and other reconnaissance missions. To adequately simulate aircraft response, the six DOF motion of the aircraft must be modeled, as well as the dynamic motion of the flapping wings. Thereby, a multibody simulation is required. In this case, each wing is modeled and simulated as a separate body, and the fuselage,


 Fig. 12 Euler roll angle (ϕ) of the wings and fuselage following the serpentine path.

vertical tail, and horizontal tail combined represent the third body. The **I** axis of each body and joint are aligned pointing out the nose of the vehicle. This is illustrated in Fig. 10. A hinge joint along the **I** axis is used between each wing and the fuselage/tail body in the simulation. Only rotation is allowed along the **I** axis of each body, resulting in the Γ matrices below.

$$\Gamma_{T_j} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \Gamma_{R_j} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (69)$$

The aircraft used in the study is configured for surveillance and reconnaissance missions and has a wingspan of 0.80 m and a mass of 0.836 kg. The aircraft is controlled with a rudder and differential

elevons. Dihedral of the wings is used as a lateral control mechanism to steer the aircraft, in place of differential elevons or ailerons. The dihedral is varied differentially and is set by changing the precone angle of the spring at the hinge joint. The rudder is used to zero any sideslip and thus perform coordinated turns. The elevons are used purely as elevators. Throttle is kept constant through the maneuver. Figure 11 shows the aircraft flying through a serpentine. In the second turn, when the aircraft is facing out of the page, it is particularly apparent how the wings are angled differently relative to the fuselage body. Figure 12 shows the Euler roll angle, ϕ , of the aircraft as it flies through the serpentine maneuver. The fuselage (dotted) roll angle lags behind the angle of the wings. The fuselage also initially rolls the opposite direction, which is due to the moments imparted on it by the springs at the hinge joints.

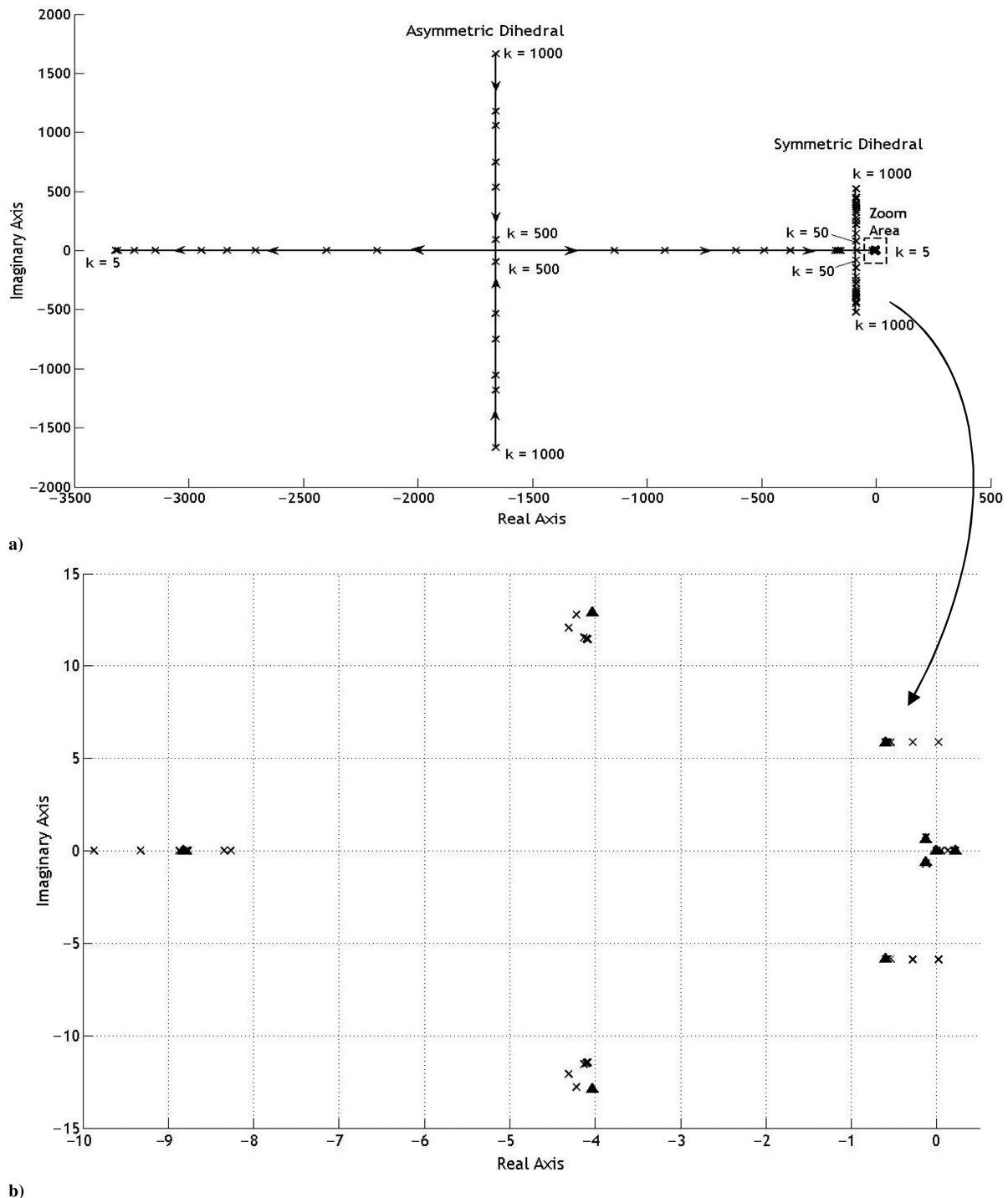


Fig. 13 a) Root locus of the articulated aircraft and b) zoom of the marked box in a) showing the low frequency eigenvalues. ▲ = rigid aircraft; x = articulated aircraft.

The translational and rotational constraint errors of each joint are measured for both the rigid and articulated case through the maneuver. The translational constraint error for both joints in both cases is very small, on the order of 10^{-13} , with a maximum error on the order of $2e10^{-10}$, so the error is being driven to zero in the constraint controller. The rotational errors are also minimal, generally on the order of 10^{-15} with a maximum error of $1e10^{-11}$. Thus, the rotational constraints are satisfied in both cases, and the constraint controller performs well.

As described in Sec. II.I, linear models of the unconstrained system can be constructed from the constrained system simulation. The constrained simulation is run until a trimmed state is reached. Following Sec. II.I, a linear model is created about this trim state. The unconstrained state matrix that is calculated can then be used to analyze the stability of the system. In particular, the eigenvalues of the system can be tracked as important parameters of the system are changed, creating a root locus plot. This depicts how the modes of the system change. Figure 13 shows the root locus of the articulated aircraft as the stiffness of the rotational springs at the hinge joints varies from 5 to 1000 N-m/rad. The lower plot of Fig. 13 zooms in on the low frequency eigenvalues. It can be seen that as the spring stiffness is increased, the modes move to match those of the rigid aircraft. In the upper plot, there are also two distinct modes that correspond to symmetric deflection of the wings and asymmetric deflection of the wings.

IV. Conclusions

A handy method for formulating flight dynamic equations of motion for air vehicle configurations that can be adequately described by a collection of rigid bodies connected by a set of joints is considered. The multibody flight dynamic simulation method leverages standard rigid six degrees of freedom aircraft flight dynamic modeling and employs it as the basic kernel from which the overall system simulation is constructed. All joints create a set of constraint forces and moments that connect the system together in an appropriate manner. In the formulation, these constraint forces and moments are treated as external loads that properly glue the system together. The glue code is based on satisfying all motion constraints using a nonlinear controller that computes the constraint loads.

In this paper, the method is applied to simulate three different multibody aerospace vehicles. The first is a projectile that uses an internal translating mass for lateral control. Through this application, it was noted that the computation of the glue code constraint loads dominates the overall run time for the simulation. Moreover, partial derivatives of the constraint equation time derivative can be computed either analytically or numerically with numerical derivatives substantially increasing run time. It is shown that the simulation can be substantially accelerated by only occasionally updating the constraint loads during simulation, when a balance between run time and constraint equation errors must be struck. An articulated wing aircraft simulation is another example. In this case, the wings are hinged at the root, and the aircraft is controlled through a serpentine maneuver by varying the wing angle. Through this application, the procedure to form linear models of the system is demonstrated, by appropriately considering the constraint equations and eliminating extra degrees of freedom, and the resulting eigenanalysis is presented. Another application simulated is an airdrop system consisting of a parafoil connected to a payload at a single confluence point, represented as a gimbal joint. In this example, both a feedback linearization and a sliding mode controller algorithm were shown to work well. The method presented in this paper was also compared with an unconstrained Newton–Euler method for this application with virtually identical results. Although the minimal coordinates method ran expectedly faster, this method does not require complex derivations or advanced knowledge of multibody dynamic methods to obtain the equations of motion and can be readily employed for a generic N body system with miscellaneous joint connections. The overall multibody flight dynamic simulation method should prove useful to the flight mechanics community because it introduces some straightforward modifications to standard rigid six degrees of

freedom modeling to permit flight simulation of multibody aircraft configurations.

Appendix: Partial Derivatives of \dot{E}

$$\begin{aligned} \dot{E}_{T_j}^* = & \Gamma_{T_j}^T T_{PJ_j} \left[\begin{array}{c} u_{P_j} \\ v_{P_j} \\ w_{P_j} \end{array} \right] - T_{P_j} T_{C_j}^T \left[\begin{array}{c} u_{C_j} \\ v_{C_j} \\ w_{C_j} \end{array} \right] \\ & - S_{\omega_{P_j}} T_{P_j} \left[\begin{array}{c} x_{P_j} - x_{C_j} \\ y_{P_j} - y_{C_j} \\ z_{P_j} - z_{C_j} \end{array} \right] + S_{\omega_{P_j}} T_{P_j} T_{C_j}^T \left[\begin{array}{c} \Delta X_{C_j} \\ \Delta Y_{C_j} \\ \Delta Z_{C_j} \end{array} \right] \\ & - T_{P_j} T_{C_j}^T S_{\omega_{C_j}} \left[\begin{array}{c} \Delta X_{C_j} \\ \Delta Y_{C_j} \\ \Delta Z_{C_j} \end{array} \right] \end{aligned}$$

$$S_{\omega_{P_j}} = \begin{bmatrix} 0 & -r_{P_j} & q_{P_j} \\ r_{P_j} & 0 & -p_{P_j} \\ -q_{P_j} & p_{P_j} & 0 \end{bmatrix}$$

$$S_{\omega_{C_j}} = \begin{bmatrix} 0 & -r_{C_j} & q_{C_j} \\ r_{C_j} & 0 & -p_{C_j} \\ -q_{C_j} & p_{C_j} & 0 \end{bmatrix}$$

$$X_{YZ_{P_j}} = [x_{P_j} \ y_{P_j} \ z_{P_j}]^T \quad X_{YZ_{C_j}} = [x_{C_j} \ y_{C_j} \ z_{C_j}]^T$$

$$\frac{\partial \dot{E}_{T_j}^*}{\partial X_{YZ_{P_j}}} = -\Gamma_{T_j}^T T_{PJ_j} S_{\omega_{P_j}} T_{P_j} \quad \frac{\partial \dot{E}_{T_j}^*}{\partial X_{YZ_{C_j}}} = \Gamma_{T_j}^T T_{PJ_j} S_{\omega_{P_j}} T_{P_j}$$

$$\frac{\partial \dot{E}_{T_j}^*}{\partial Q_{P_j}} = \left[\begin{array}{c} \frac{\partial \dot{E}_{T_j}^*}{\partial q_{0_{P_j}}} \quad \frac{\partial \dot{E}_{T_j}^*}{\partial q_{1_{P_j}}} \quad \frac{\partial \dot{E}_{T_j}^*}{\partial q_{2_{P_j}}} \quad \frac{\partial \dot{E}_{T_j}^*}{\partial q_{3_{P_j}}} \end{array} \right]$$

$$\begin{aligned} \frac{\partial \dot{E}_{T_j}^*}{\partial q_{0_{P_j}}} = & \Gamma_{T_j}^T T_{PJ_j} S_{\omega_{P_j}} \frac{\partial T_{P_j}}{\partial q_{0_{P_j}}} \left[T_{C_j}^T \left[\begin{array}{c} \Delta X_{C_j} \\ \Delta Y_{C_j} \\ \Delta Z_{C_j} \end{array} \right] - \left[\begin{array}{c} x_{P_j} - x_{C_j} \\ y_{P_j} - y_{C_j} \\ z_{P_j} - z_{C_j} \end{array} \right] \right] \\ & - \Gamma_{T_j}^T T_{PJ_j} \frac{\partial T_{P_j}}{\partial q_{0_{P_j}}} T_{C_j}^T \left[\begin{array}{c} u_{C_j} \\ v_{C_j} \\ w_{C_j} \end{array} \right] - S_{C_j} \left[\begin{array}{c} p_{C_j} \\ q_{C_j} \\ r_{C_j} \end{array} \right] \end{aligned}$$

$$\begin{aligned} \frac{\partial \dot{E}_{T_j}^*}{\partial q_{1_{P_j}}} = & \Gamma_{T_j}^T T_{PJ_j} S_{\omega_{P_j}} \frac{\partial T_{P_j}}{\partial q_{1_{P_j}}} \left[T_{C_j}^T \left[\begin{array}{c} \Delta X_{C_j} \\ \Delta Y_{C_j} \\ \Delta Z_{C_j} \end{array} \right] - \left[\begin{array}{c} x_{P_j} - x_{C_j} \\ y_{P_j} - y_{C_j} \\ z_{P_j} - z_{C_j} \end{array} \right] \right] \\ & - \Gamma_{T_j}^T T_{PJ_j} \frac{\partial T_{P_j}}{\partial q_{1_{P_j}}} T_{C_j}^T \left[\begin{array}{c} u_{C_j} \\ v_{C_j} \\ w_{C_j} \end{array} \right] - S_{C_j} \left[\begin{array}{c} p_{C_j} \\ q_{C_j} \\ r_{C_j} \end{array} \right] \end{aligned}$$

$$\begin{aligned} \frac{\partial \dot{E}_{T_j}^*}{\partial q_{2_{P_j}}} = & \Gamma_{T_j}^T T_{PJ_j} S_{\omega_{P_j}} \frac{\partial T_{P_j}}{\partial q_{2_{P_j}}} \left[T_{C_j}^T \left[\begin{array}{c} \Delta X_{C_j} \\ \Delta Y_{C_j} \\ \Delta Z_{C_j} \end{array} \right] - \left[\begin{array}{c} x_{P_j} - x_{C_j} \\ y_{P_j} - y_{C_j} \\ z_{P_j} - z_{C_j} \end{array} \right] \right] \\ & - \Gamma_{T_j}^T T_{PJ_j} \frac{\partial T_{P_j}}{\partial q_{2_{P_j}}} T_{C_j}^T \left[\begin{array}{c} u_{C_j} \\ v_{C_j} \\ w_{C_j} \end{array} \right] - S_{C_j} \left[\begin{array}{c} p_{C_j} \\ q_{C_j} \\ r_{C_j} \end{array} \right] \end{aligned}$$

$$\frac{\partial \dot{E}_{T_j}^*}{\partial q_{3p_j}} = \Gamma_{T_j}^T T_{PJ_j} S_{\omega_{p_j}} \frac{\partial T_{P_j}}{\partial q_{3p_j}} \left[T_{C_j}^T \begin{Bmatrix} \Delta X_{C_j} \\ \Delta Y_{C_j} \\ \Delta Z_{C_j} \end{Bmatrix} - \begin{Bmatrix} x_{p_j} - x_{C_j} \\ y_{p_j} - y_{C_j} \\ z_{p_j} - z_{C_j} \end{Bmatrix} \right] \\ - \Gamma_{T_j}^T T_{PJ_j} \frac{\partial T_{P_j}}{\partial q_{3p_j}} T_{C_j}^T \left[\begin{Bmatrix} u_{C_j} \\ v_{C_j} \\ w_{C_j} \end{Bmatrix} - S_{C_j} \begin{Bmatrix} p_{C_j} \\ q_{C_j} \\ r_{C_j} \end{Bmatrix} \right]$$

$$\frac{\partial T_{P_j}}{\partial q_{0p_j}} = 2 \begin{bmatrix} q_{0p_j} & q_{3p_j} & -q_{2p_j} \\ -q_{3p_j} & q_{0p_j} & q_{1p_j} \\ q_{2p_j} & -q_{1p_j} & q_{0p_j} \end{bmatrix}$$

$$\frac{\partial T_{P_j}}{\partial q_{1p_j}} = 2 \begin{bmatrix} q_{1p_j} & q_{2p_j} & q_{3p_j} \\ q_{2p_j} & -q_{1p_j} & q_{0p_j} \\ q_{3p_j} & -q_{0p_j} & -q_{1p_j} \end{bmatrix}$$

$$\frac{\partial T_{P_j}}{\partial q_{2p_j}} = 2 \begin{bmatrix} -q_{2p_j} & q_{1p_j} & -q_{0p_j} \\ q_{1p_j} & q_{2p_j} & q_{3p_j} \\ q_{0p_j} & q_{3p_j} & -q_{2p_j} \end{bmatrix}$$

$$\frac{\partial T_{P_j}}{\partial q_{3p_j}} = 2 \begin{bmatrix} -q_{3p_j} & q_{0p_j} & q_{1p_j} \\ -q_{0p_j} & -q_{3p_j} & q_{2p_j} \\ q_{1p_j} & q_{2p_j} & q_{3p_j} \end{bmatrix}$$

$$\frac{\partial \dot{E}_{T_j}^*}{\partial Q_{C_j}} = \begin{bmatrix} \frac{\partial \dot{E}_{T_j}^*}{\partial q_{0c_j}} & \frac{\partial \dot{E}_{T_j}^*}{\partial q_{1c_j}} & \frac{\partial \dot{E}_{T_j}^*}{\partial q_{2c_j}} & \frac{\partial \dot{E}_{T_j}^*}{\partial q_{3c_j}} \end{bmatrix}$$

$$\frac{\partial \dot{E}_{T_j}^*}{\partial q_{0c_j}} = -\Gamma_{T_j}^T T_{PJ_j} T_{P_j} \frac{\partial T_{C_j}^T}{\partial q_{0c_j}} \left[\begin{Bmatrix} u_{C_j} \\ v_{C_j} \\ w_{C_j} \end{Bmatrix} - S_{C_j} \begin{Bmatrix} p_{C_j} \\ q_{C_j} \\ r_{C_j} \end{Bmatrix} \right] \\ + \Gamma_{T_j}^T T_{PJ_j} S_{\omega_{p_j}} T_P \frac{\partial T_{C_j}^T}{\partial q_{0c_j}} \begin{Bmatrix} \Delta X_{C_j} \\ \Delta Y_{C_j} \\ \Delta Z_{C_j} \end{Bmatrix}$$

$$\frac{\partial \dot{E}_{T_j}^*}{\partial q_{1c_j}} = -\Gamma_{T_j}^T T_{PJ_j} T_{P_j} \frac{\partial T_{C_j}^T}{\partial q_{1c_j}} \left[\begin{Bmatrix} u_{C_j} \\ v_{C_j} \\ w_{C_j} \end{Bmatrix} - S_{C_j} \begin{Bmatrix} p_{C_j} \\ q_{C_j} \\ r_{C_j} \end{Bmatrix} \right] \\ + \Gamma_{T_j}^T T_{PJ_j} S_{\omega_{p_j}} T_P \frac{\partial T_{C_j}^T}{\partial q_{1c_j}} \begin{Bmatrix} \Delta X_{C_j} \\ \Delta Y_{C_j} \\ \Delta Z_{C_j} \end{Bmatrix}$$

$$\frac{\partial \dot{E}_{T_j}^*}{\partial q_{2c_j}} = -\Gamma_{T_j}^T T_{PJ_j} T_{P_j} \frac{\partial T_{C_j}^T}{\partial q_{2c_j}} \left[\begin{Bmatrix} u_{C_j} \\ v_{C_j} \\ w_{C_j} \end{Bmatrix} - S_{C_j} \begin{Bmatrix} p_{C_j} \\ q_{C_j} \\ r_{C_j} \end{Bmatrix} \right] \\ + \Gamma_{T_j}^T T_{PJ_j} S_{\omega_{p_j}} T_P \frac{\partial T_{C_j}^T}{\partial q_{2c_j}} \begin{Bmatrix} \Delta X_{C_j} \\ \Delta Y_{C_j} \\ \Delta Z_{C_j} \end{Bmatrix}$$

$$\frac{\partial \dot{E}_{T_j}^*}{\partial q_{3c_j}} = -\Gamma_{T_j}^T T_{PJ_j} T_{P_j} \frac{\partial T_{C_j}^T}{\partial q_{3c_j}} \left[\begin{Bmatrix} u_{C_j} \\ v_{C_j} \\ w_{C_j} \end{Bmatrix} - S_{C_j} \begin{Bmatrix} p_{C_j} \\ q_{C_j} \\ r_{C_j} \end{Bmatrix} \right] \\ + \Gamma_{T_j}^T T_{PJ_j} S_{\omega_{p_j}} T_P \frac{\partial T_{C_j}^T}{\partial q_{3c_j}} \begin{Bmatrix} \Delta X_{C_j} \\ \Delta Y_{C_j} \\ \Delta Z_{C_j} \end{Bmatrix}$$

$$\frac{\partial T_{C_j}}{\partial q_{0c_j}} = 2 \begin{bmatrix} q_{0c_j} & q_{3c_j} & -q_{2c_j} \\ -q_{3c_j} & q_{0c_j} & q_{1c_j} \\ q_{2c_j} & -q_{1c_j} & q_{0c_j} \end{bmatrix}$$

$$\frac{\partial T_{C_j}}{\partial q_{1c_j}} = 2 \begin{bmatrix} q_{1c_j} & q_{2c_j} & q_{3c_j} \\ q_{2c_j} & -q_{1c_j} & q_{0c_j} \\ q_{3c_j} & -q_{0c_j} & -q_{1c_j} \end{bmatrix}$$

$$\frac{\partial T_{C_j}}{\partial q_{2c_j}} = 2 \begin{bmatrix} -q_{2c_j} & q_{1c_j} & -q_{0c_j} \\ q_{1c_j} & q_{2c_j} & q_{3c_j} \\ q_{0c_j} & q_{3c_j} & -q_{2c_j} \end{bmatrix}$$

$$\frac{\partial T_{C_j}}{\partial q_{3c_j}} = 2 \begin{bmatrix} -q_{3c_j} & q_{0c_j} & q_{1c_j} \\ -q_{0c_j} & -q_{3c_j} & q_{2c_j} \\ q_{1c_j} & q_{2c_j} & q_{3c_j} \end{bmatrix}$$

$$U_{VW_{P_j}} = [u_{P_j} \quad v_{P_j} \quad w_{P_j}]^T \quad U_{VW_{C_j}} = [u_{C_j} \quad v_{C_j} \quad w_{C_j}]^T$$

$$\frac{\partial \dot{E}_{T_j}^*}{\partial U_{VW_{P_j}}} = \Gamma_{T_j}^T T_{PJ_j} \quad \frac{\partial \dot{E}_{T_j}^*}{\partial U_{VW_{C_j}}} = -\Gamma_{T_j}^T T_{PJ_j} T_{P_j} T_{C_j}^T$$

$$\frac{\partial \dot{E}_{T_j}^*}{\partial \Omega_{P_j}} = \begin{bmatrix} \frac{\partial \dot{E}_{T_j}^*}{\partial p_{P_j}} & \frac{\partial \dot{E}_{T_j}^*}{\partial q_{P_j}} & \frac{\partial \dot{E}_{T_j}^*}{\partial r_{P_j}} \end{bmatrix}$$

$$\frac{\partial \dot{E}_{T_j}^*}{\partial \Omega_{C_j}} = \Gamma_{T_j}^T T_{PJ_j} T_{P_j} T_{C_j}^T S_{C_j}$$

$$\frac{\partial S_{\omega_{P_j}}}{\partial p_{P_j}} = \frac{\partial S_{\omega_{C_j}}}{\partial p_{C_j}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\frac{\partial S_{\omega_{P_j}}}{\partial q_{P_j}} = \frac{\partial S_{\omega_{C_j}}}{\partial q_{C_j}} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial S_{\omega_{P_j}}}{\partial r_{P_j}} = \frac{\partial S_{\omega_{C_j}}}{\partial r_{C_j}} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial \dot{E}_{T_j}^*}{\partial p_{P_j}} = \Gamma_{T_j}^T T_{PJ_j} \frac{\partial S_{\omega_{P_j}}}{\partial p_{P_j}} T_{P_j} \left[T_{C_j}^T \begin{Bmatrix} \Delta X_{C_j} \\ \Delta Y_{C_j} \\ \Delta Z_{C_j} \end{Bmatrix} - \begin{Bmatrix} x_{P_j} - x_{C_j} \\ y_{P_j} - y_{C_j} \\ z_{P_j} - z_{C_j} \end{Bmatrix} \right]$$

$$\frac{\partial \dot{E}_{T_j}^*}{\partial q_{P_j}} = \Gamma_{T_j}^T T_{PJ_j} \frac{\partial S_{\omega_{P_j}}}{\partial q_{P_j}} T_{P_j} \left[T_{C_j}^T \begin{Bmatrix} \Delta X_{C_j} \\ \Delta Y_{C_j} \\ \Delta Z_{C_j} \end{Bmatrix} - \begin{Bmatrix} x_{P_j} - x_{C_j} \\ y_{P_j} - y_{C_j} \\ z_{P_j} - z_{C_j} \end{Bmatrix} \right]$$

$$\frac{\partial \dot{E}_{T_j}^*}{\partial r_{P_j}} = \Gamma_{T_j}^T T_{PJ_j} \frac{\partial S_{\omega_{P_j}}}{\partial r_{P_j}} T_{P_j} \left[T_{C_j}^T \begin{Bmatrix} \Delta X_{C_j} \\ \Delta Y_{C_j} \\ \Delta Z_{C_j} \end{Bmatrix} - \begin{Bmatrix} x_{P_j} - x_{C_j} \\ y_{P_j} - y_{C_j} \\ z_{P_j} - z_{C_j} \end{Bmatrix} \right]$$

$$\Omega_{C_j} = [p_{C_j} \quad q_{C_j} \quad r_{C_j}]^T$$

$$\dot{E}_{R_j}^* = -\Phi_j T_{PJ_j} (S_{\omega_{P_j}} T_{P_j} T_{C_j}^T - T_{P_j} T_{C_j}^T S_{\omega_{C_j}}) T_{C_j}^T \Psi_j^T$$

$$\frac{\partial \dot{E}_{R_j}^*}{\partial U_{VW_{P_j}}} = \frac{\partial \dot{E}_{R_j}^*}{\partial U_{VW_{C_j}}} = \frac{\partial \dot{E}_{R_j}^*}{\partial X_{YZ_{P_j}}} = \frac{\partial \dot{E}_{R_j}^*}{\partial X_{YZ_{C_j}}} = 0$$

$$\frac{\partial \dot{E}_{R_j}^*}{\partial Q_{P_j}} = \begin{bmatrix} \frac{\partial \dot{E}_{R_j}^*}{\partial q_{0_{P_j}}} & \frac{\partial \dot{E}_{R_j}^*}{\partial q_{1_{P_j}}} & \frac{\partial \dot{E}_{R_j}^*}{\partial q_{2_{P_j}}} & \frac{\partial \dot{E}_{R_j}^*}{\partial q_{3_{P_j}}} \end{bmatrix}$$

$$\frac{\partial \dot{E}_{R_j}^*}{\partial q_{0_{P_j}}} = -\Phi_j T_{PJ_j} \left(S_{\omega_{P_j}} \frac{\partial T_{P_j}}{\partial q_{0_{P_j}}} T_{C_j}^T - \frac{\partial T_{P_j}}{\partial q_{0_{P_j}}} T_{C_j}^T S_{\omega_{C_j}} \right) T_{C_j}^T \Psi_j^T$$

$$\frac{\partial \dot{E}_{R_j}^*}{\partial q_{1_{P_j}}} = -\Phi_j T_{PJ_j} \left(S_{\omega_{P_j}} \frac{\partial T_{P_j}}{\partial q_{1_{P_j}}} T_{C_j}^T - \frac{\partial T_{P_j}}{\partial q_{1_{P_j}}} T_{C_j}^T S_{\omega_{C_j}} \right) T_{C_j}^T \Psi_j^T$$

$$\frac{\partial \dot{E}_{R_j}^*}{\partial q_{2_{P_j}}} = -\Phi_j T_{PJ_j} \left(S_{\omega_{P_j}} \frac{\partial T_{P_j}}{\partial q_{2_{P_j}}} T_{C_j}^T - \frac{\partial T_{P_j}}{\partial q_{2_{P_j}}} T_{C_j}^T S_{\omega_{C_j}} \right) T_{C_j}^T \Psi_j^T$$

$$\frac{\partial \dot{E}_{R_j}^*}{\partial q_{3_{P_j}}} = -\Phi_j T_{PJ_j} \left(S_{\omega_{P_j}} \frac{\partial T_{P_j}}{\partial q_{3_{P_j}}} T_{C_j}^T - \frac{\partial T_{P_j}}{\partial q_{3_{P_j}}} T_{C_j}^T S_{\omega_{C_j}} \right) T_{C_j}^T \Psi_j^T$$

$$\frac{\partial \dot{E}_{R_j}^*}{\partial Q_{C_j}} = \begin{bmatrix} \frac{\partial \dot{E}_{R_j}^*}{\partial q_{0_{C_j}}} & \frac{\partial \dot{E}_{R_j}^*}{\partial q_{1_{C_j}}} & \frac{\partial \dot{E}_{R_j}^*}{\partial q_{2_{C_j}}} & \frac{\partial \dot{E}_{R_j}^*}{\partial q_{3_{C_j}}} \end{bmatrix}$$

$$\frac{\partial \dot{E}_{R_j}^*}{\partial q_{0_{C_j}}} = -\Phi_j T_{PJ_j} \left(S_{\omega_{P_j}} T_{P_j} \frac{\partial T_{C_j}^T}{\partial q_{0_{C_j}}} - T_{P_j} \frac{\partial T_{C_j}^T}{\partial q_{0_{C_j}}} S_{\omega_{C_j}} \right) T_{C_j}^T \Psi_j^T$$

$$\frac{\partial \dot{E}_{R_j}^*}{\partial q_{1_{C_j}}} = -\Phi_j T_{PJ_j} \left(S_{\omega_{P_j}} T_{P_j} \frac{\partial T_{C_j}^T}{\partial q_{1_{C_j}}} - T_{P_j} \frac{\partial T_{C_j}^T}{\partial q_{1_{C_j}}} S_{\omega_{C_j}} \right) T_{C_j}^T \Psi_j^T$$

$$\frac{\partial \dot{E}_{R_j}^*}{\partial q_{2_{C_j}}} = -\Phi_j T_{PJ_j} \left(S_{\omega_{P_j}} T_{P_j} \frac{\partial T_{C_j}^T}{\partial q_{2_{C_j}}} - T_{P_j} \frac{\partial T_{C_j}^T}{\partial q_{2_{C_j}}} S_{\omega_{C_j}} \right) T_{C_j}^T \Psi_j^T$$

$$\frac{\partial \dot{E}_{R_j}^*}{\partial q_{3_{C_j}}} = -\Phi_j T_{PJ_j} \left(S_{\omega_{P_j}} T_{P_j} \frac{\partial T_{C_j}^T}{\partial q_{3_{C_j}}} - T_{P_j} \frac{\partial T_{C_j}^T}{\partial q_{3_{C_j}}} S_{\omega_{C_j}} \right) T_{C_j}^T \Psi_j^T$$

$$\frac{\partial \dot{E}_{R_j}^*}{\partial \Omega_{P_j}} = \begin{bmatrix} \frac{\partial \dot{E}_{R_j}^*}{\partial p_{P_j}} & \frac{\partial \dot{E}_{R_j}^*}{\partial q_{P_j}} & \frac{\partial \dot{E}_{R_j}^*}{\partial r_{P_j}} \end{bmatrix}$$

$$\frac{\partial \dot{E}_{R_j}^*}{\partial p_{P_j}} = -\Phi_j T_{PJ_j} \frac{\partial S_{\omega_{P_j}}}{\partial p_{P_j}} T_{P_j} T_{C_j}^T T_{C_j}^T \Psi_j^T$$

$$\frac{\partial \dot{E}_{R_j}^*}{\partial q_{P_j}} = -\Phi_j T_{PJ_j} \frac{\partial S_{\omega_{P_j}}}{\partial q_{P_j}} T_{P_j} T_{C_j}^T T_{C_j}^T \Psi_j^T$$

$$\frac{\partial \dot{E}_{R_j}^*}{\partial r_{P_j}} = -\Phi_j T_{PJ_j} \frac{\partial S_{\omega_{P_j}}}{\partial r_{P_j}} T_{P_j} T_{C_j}^T T_{C_j}^T \Psi_j^T$$

$$\frac{\partial \dot{E}_{R_j}^*}{\partial \Omega_{C_j}} = \begin{bmatrix} \frac{\partial \dot{E}_{R_j}^*}{\partial p_{C_j}} & \frac{\partial \dot{E}_{R_j}^*}{\partial q_{C_j}} & \frac{\partial \dot{E}_{R_j}^*}{\partial r_{C_j}} \end{bmatrix}$$

$$\frac{\partial \dot{E}_{R_j}^*}{\partial p_{C_j}} = \Phi_j T_{PJ_j} T_{P_j} T_{C_j}^T \frac{\partial S_{\omega_{C_j}}}{\partial p_{C_j}} T_{C_j}^T \Psi_j^T$$

$$\frac{\partial \dot{E}_{R_j}^*}{\partial q_{C_j}} = \Phi_j T_{PJ_j} T_{P_j} T_{C_j}^T \frac{\partial S_{\omega_{C_j}}}{\partial q_{C_j}} T_{C_j}^T \Psi_j^T$$

$$\frac{\partial \dot{E}_{R_j}^*}{\partial r_{C_j}} = \Phi_j T_{PJ_j} T_{P_j} T_{C_j}^T \frac{\partial S_{\omega_{C_j}}}{\partial r_{C_j}} T_{C_j}^T \Psi_j^T$$

Acknowledgments

The authors acknowledge Kyle French for his contributions in developing the paper. The authors also acknowledge Blaine Costello and Mariel Hiemstra, who generated the application CAD drawings.

References

- [1] Frost, G., and Costello, M., "Control Authority of a Projectile Equipped with an Internal Unbalanced Part," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 128, No. 4, 2006, pp. 1005–1012. doi:10.1115/1.2363205
- [2] Rogers, J., and Costello, M., "Control Authority of a Projectile Equipped with a Controllable Internal Translating Mass," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, 2008, pp. 1323–1333. doi:10.2514/1.33961
- [3] Rogers, J., and Costello, M., "A Variable Stability Projectile Using an Internal Moving Mass," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 223, No. 7, 2009, pp. 927–938.
- [4] Rogers, J., and Costello, M., "Cantilever Beam Design for Projectile Internal Moving Mass Systems," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 131, No. 5, 2009, pp. 1–11.
- [5] Costello, M., and Peterson, A., "Linear Theory of a Dual Spin Projectile in Atmospheric Flight," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 5, 2000, pp. 789–797. doi:10.2514/2.4639
- [6] Burchett, B., Peterson, A., and Costello, M., "Prediction of Swerving Motion of a Dual-Spin Projectile with Lateral Pulse Jets in Atmospheric Flight," *Mathematical and Computer Modelling*, Vol. 35, Nos. 7–8, 2002, pp. 821–834.
- [7] Costello, M., and Agarwalla, R., "Improved Dispersion of a Fin-Stabilized Projectile Using a Passive Movable Nose," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 5, pp. 900–903, 2000; doi:10.2514/2.4625Errata: Vol. 25, No. 2, 2002, p. 414. doi:10.2514/2.4898
- [8] Slegers, N., and Costello, M., "Aspects of Control for a Parafoil and Payload System," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 6, 2003, pp. 898–905. doi:10.2514/2.6933
- [9] Slegers, N., and Costello, M., "Model Predictive Control of a Parafoil and Payload System," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, 2005, pp. 816–821. doi:10.2514/1.12251
- [10] Slegers, N., Beyer, E., and Costello, M., "Use of Dynamic Incidence Angle for Glide Slope Control of Autonomous Parafoils," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 585–596. doi:10.2514/1.32099
- [11] Stewart, K., Blackburn, K., Wagener, J., Czabaranek, J., and Abate, G., "Development and Initial Flight Tests of a Single-Jointed Articulated-Wing Micro Air Vehicle," *AIAA Atmospheric Flight Mechanics Conference*, AIAA 2008-6708, Honolulu, HI, Aug. 2008.
- [12] Garcia de Jalon, J., and Bayo, E., *Kinematic and Dynamic Simulation of Multibody Systems*, Springer-Verlag, New York, 1994, pp. 7, 16–63, 156–196.
- [13] Shabana, A. A., *Dynamics of Multibody Systems*, 2nd ed., Cambridge Univ. Press, Cambridge, U.K., 1998, pp. 1–3, 19–22.

- [14] Roberson, R. E., and Schwertassek, R., *Dynamics of Multibody Systems*, Springer, Berlin, 1988, pp. 176–177, 181–183, 245–254.
- [15] Eberhard, P., and Schliehlen, W., “Computational Dynamics of Multibody Systems: History, Formalisms, and Applications,” *Journal of Computational and Nonlinear Dynamics*, Vol. 1, No. 1, 2006, pp. 3–12.
doi:10.1115/1.1961875
- [16] Huston, R. L., “Multibody Dynamics: Modeling and Analysis Methods,” *Applied Mechanics Reviews*, Vol. 44, 1991, pp. 109–117.
doi:10.1115/1.3119496
- [17] Chiou, J., and Wu, S., “Constraint Violation Stabilization Using Input-Output Feedback Linearization in Multibody Dynamic Analysis,” *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 222–228.
doi:10.2514/2.4246
- [18] Vlasenko, D., and Kasper, R., “A New Software Approach for the Simulation of Multibody Dynamics,” *Journal of Computational and Nonlinear Dynamics*, Vol. 2, No. 3, 2007, pp. 274–278.
doi:10.1115/1.2734182
- [19] Park, K. C., Chiou, J. C., and Downer, J. D., “Explicit-Implicit Staggered Procedure for Multibody Dynamics Analysis,” *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 3, 1990, pp. 562–570.
doi:10.2514/3.25370
- [20] Kurdila, A., Papastavidris, J. G., and Kamat, M. P., “Role of Maggi’s Equations in Computational Methods for Constrained Multibody Systems,” *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 1, 1990, pp. 113–120.
doi:10.2514/3.20524
- [21] Garcia de Jalon, J., Unda, J., and Avello, A., “Natural Coordinates for the Computer Analysis of Multibody Systems,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 56, 1986, pp. 309–327.
doi:10.1016/0045-7825(86)90044-7
- [22] Park, K. C., and Chiou, J. C., “Stabilization of Computational Procedures for Constrained Dynamical Systems,” *Journal of Guidance, Control, and Dynamics*, Vol. 11, No. 4, 1988, pp. 365–370.
doi:10.2514/3.20320
- [23] Bauchau, O. A., and Laulusa, A., “Review of Contemporary Approaches for Constraint Enforcement in Multibody Systems,” *Journal of Computational and Nonlinear Dynamics*, Vol. 3, No. 1, Jan. 2008, pp. 1–8.
- [24] Ginsberg, J., *Advanced Engineering Dynamics*, 2nd ed., Cambridge Univ. Press, New York, 1995, pp. 163–220.
- [25] Kane, T., and Levinson, H., *Dynamics: Theory and Applications*, McGraw Hill Company, New York, 1985, pp. 158–186.
- [26] Baruh, H., *Analytical Dynamics*, McGraw Hill Company, Boston, MA, 1999, pp. 216–260.
- [27] Kuipers, J., *Quaternions and Rotation Sequences*, Princeton Univ. Press, Princeton, NJ, 1999, pp. 77–91, 166–168.
- [28] Slotine, J. J., and Li, W., *Applied Nonlinear Control*, Prentice Hall Inc., Englewood Cliffs, NJ, 1991, pp. 207–266, 276–303.