Research Article

Static Analysis of Large-Scale Multibody System Using Joint Coordinates and Spatial Algebra Operator

Mohamed A. Omar

Mechanical Engineering Department, Taibah University, Almadinah Almonawwarah 42353, Saudi Arabia

Correspondence should be addressed to Mohamed A. Omar; momar@taibahu.edu.sa

Received 15 March 2014; Accepted 28 March 2014; Published 19 June 2014

Academic Editor: Belal F. Yousif

Copyright © 2014 Mohamed A. Omar. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Initial transient oscillations inhibited in the dynamic simulations responses of multibody systems can lead to inaccurate results, unrealistic load prediction, or simulation failure. These transients could result from incompatible initial conditions, initial constraints violation, and inadequate kinematic assembly. Performing static equilibrium analysis before the dynamic simulation can eliminate these transients and lead to stable simulation. Most exiting multibody formulations determine the static equilibrium position by minimizing the system potential energy. This paper presents a new general purpose approach for solving the static equilibrium in large-scale articulated multibody. The proposed approach introduces an energy drainage mechanism based on Baumgarte constraint stabilization approach to determine the static equilibrium position. The spatial algebra operator is used to express the kinematic and dynamic equations of the closed-loop multibody system. The proposed multibody system formulation utilizes the joint coordinates and modal elastic coordinates as the system generalized coordinates. The recursive nonlinear equations of motion are formulated using the Cartesian coordinates and the joint coordinates to form an augmented set of differential algebraic equations. Then system connectivity matrix is derived from the system topological relations and used to project the Cartesian quantities into the joint subspace leading to minimum set of differential equations.

1. Introduction

Virtual development procedures became the most economical venue in product design and optimization including earth-moving equipment and automotive systems. These systems are typically large and complex and composed of heterogeneous physical subsystems. Simulation of such systems may require multidomain modeling, for example, rigid and flexible bodies, nonlinear contact and friction force modules, terrain interaction, electrical and hydraulic subsystems, control systems, and nonholonomic constraints. Accurate multibody solver will be required in the heart of the analysis of such systems. As per Giampiero and Manfredi [1], the computation of static equilibrium is often the first step in the analysis of multibody system models. The static equilibrium position defines the working point for the linearization of the system equation of motion and provides initial values for the dynamical simulation.

Large earth-moving equipment, agricultural machinery, tracked vehicles, tractors, bulldozers, motor graders, wheeled

loaders, and forestry machines are some of the equipment that may work on sloped terrain. Stability assessment of such machinery on the slopes under different loading conditions and different articulation configurations requires accurate modeling of the machine and its interaction with the terrain. The behavior of existing terrain models, in most cases [2–5], is highly nonlinear and history dependent. Performing static equilibrium is very crucial step for accurate linear analysis and stability assessment of the multibody systems.

Another example would be when the multibody dynamics approach is used to model and simulate the startup and shut down of reciprocating engines and turbines which are supported by journal bearings. Journal bearings are typically nonlinear components and their dynamic characteristics (stiffness, damping, etc.) are history dependent [6, 7]. In such systems, the crankshaft and the rotor could be modeled as flexible (deformable bodies). The static equilibrium position of rotor center with respect to the bearing geometric center needs to be accurately determined in order to perform transient simulation or normal mode analysis. When

the multibody system contains contacting bodies that may undergo slipping, static equilibrium cannot be obtained from force balance. In this case, the traditional static equilibrium solution will not converge to the true static equilibrium position.

In general, analysis and simulation scenarios of dynamic multibody systems could be summarized as follows:

- (1) performing transient dynamic simulation of typical machine working cycle or operation;
- (2) performing eigenvalue analysis to calculate the natural frequencies and normal modes and assess the system stability;
- (3) determining the static equilibrium of an articulated system that may include rigid and flexible bodies connected with joints.

Three traditional approaches have been used to formulate the multibody system equations of motion: the Cartesian body coordinates (absolute coordinates), the joint coordinates, and velocity transformation method. The Cartesian body coordinates formulations are very popular and reported to be a simpler method to construct the equations of motion leading to a large set of differential algebraic equations [8, 9]. The configuration of a rigid body is described by a set of translational and rotational coordinates. Algebraic constraints are introduced to represent kinematic joints connecting bodies and then the Lagrange multiplier technique is used to describe joint reaction forces. The system of differential algebraic equations has to be solved simultaneously. Newton-Raphson iterations or similar techniques could be used to satisfy the applied constraints. Kinematic assembly of the system is necessary for starting a successful simulation.

Featherstone [10, 11] used spatial vectors to study the dynamics of articulated bodies. Featherstone and Orin [12] and Featherstone [13] presented an efficient approach for utilizing spatial algebra to model multibody systems and efficiently factor the inertia matrix for rigid body systems. Wehage and Haug [14] used a similar approach to develop set of automated procedures for robust and efficient solution of overconstrained multibody dynamics. Wehage and Belczynski [15] proposed structuring the kinematic and dynamic equations into block matrix structure and developed procedures to enable real-time simulation of multibody system. Rodriguez et al. [16, 17] presented a spatial operator based on the spatial algebra for developing multibody dynamic equations of motion. Determining the static equilibrium position using these formulations was not explicitly explained.

Traditional approach of solving for the static equilibrium is based on minimizing the potential energy of the system [1, 18]. This approach requires explicit expression for potential energy of all the bodies and the force elements in the system. This could be practically impossible for the abovementioned heterogeneous system simulations. Other approaches [19, 20] assume the system will be under static equilibrium when the first and second derivatives of the system generalized coordinates are set to zero. The differential equations of motion of the system then become a set of algebraic equations that could be solved along with the constraint equations. The

resulting solution yields a set of initial values for the system states. Linear modes analysis could then be performed by carrying out Taylor expansion while only the linear terms are retained [1, 19]. In the abovementioned approaches, false equilibrium position could be reported. Also, violation of the constraints in the velocity level and acceleration level may exist.

Recently, Yang et al. [19] presented a general approach for solving the eigenvalue problem after calculating static equilibrium. The approach is similar to many existing solution approaches and could be summarized by [19] as follows. The system equations of motion could be derived from

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{x}}} \right) + \left(\frac{\partial L}{\partial \mathbf{x}} \right) = \left(\frac{\partial \Phi}{\partial \mathbf{x}} \right)^T \boldsymbol{\lambda} + \mathbf{f}, \quad (1)$$

$$\Phi(\mathbf{x}) = \mathbf{0},$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector of general coordinates, $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the vector of Lagrange multipliers, L is the Lagrange function associated with $(\mathbf{x}, \dot{\mathbf{x}})$, $\mathbf{f}$ is the vector of nonconservative forces which are function of $(\mathbf{x}, \dot{\mathbf{x}}, t)$, Φ is the vector of constraints associated with $(\mathbf{x})$, and $(\partial \Phi / \partial \mathbf{x})^T \boldsymbol{\lambda}$ is the vector of generalized constraint forces.

The general approach to solve for the static equilibrium position starts by writing the objective function as follows:

$$\mathbf{F}(\ddot{\mathbf{x}}, \dot{\mathbf{x}}, \mathbf{x}, t) = \left(-\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{x}}} \right) + \frac{\partial L}{\partial \mathbf{x}} + \mathbf{f} \right)(\ddot{\mathbf{x}}, \dot{\mathbf{x}}, \mathbf{x}, t), \quad (2)$$

$$\mathbf{G}(\mathbf{x}) = \left(\frac{\partial \Phi(\mathbf{x})}{\partial \mathbf{x}} \right)^T.$$

Then, the system equations could be written as

$$\mathbf{F}(\ddot{\mathbf{x}}, \dot{\mathbf{x}}, \mathbf{x}, t) + \mathbf{G}(\mathbf{x}) \boldsymbol{\lambda} = \mathbf{0}, \quad (3)$$

$$\Phi(\mathbf{x}) = \mathbf{0}.$$

The equilibrium state $(\mathbf{x}_0, \boldsymbol{\lambda}_0, t_0)$ satisfies the following conditions:

$$\mathbf{F}(0, 0, \mathbf{x}_0, t_0) + \mathbf{G}(\mathbf{x}_0) \boldsymbol{\lambda}_0 = \mathbf{0}, \quad (4)$$

$$\Phi(\mathbf{x}_0) = \mathbf{0}.$$

With a small vibration about the equilibrium position $(\delta \ddot{\mathbf{x}}, \delta \dot{\mathbf{x}}, \mathbf{x}_0 + \delta \mathbf{x}, \boldsymbol{\lambda}_0 + \delta \boldsymbol{\lambda}, t)$, then

$$\mathbf{F}(\delta \ddot{\mathbf{x}}, \delta \dot{\mathbf{x}}, \mathbf{x}_0 + \delta \mathbf{x}, t) + \mathbf{G}(\mathbf{x}_0 + \delta \mathbf{x})(\boldsymbol{\lambda}_0 + \delta \boldsymbol{\lambda}) = \mathbf{0}, \quad (5)$$

$$\Phi(\mathbf{x}_0 + \delta \mathbf{x}) = \mathbf{0}.$$

Using Taylor expansion and reserving the linear terms, we get

$$\widehat{\mathbf{M}}(\mathbf{x}_0) \delta \ddot{\mathbf{y}} + \widehat{\mathbf{C}}(\mathbf{x}_0) \delta \dot{\mathbf{y}} + \widehat{\mathbf{K}}(\mathbf{x}_0, \boldsymbol{\lambda}_0) \delta \mathbf{y} = \mathbf{0}, \quad (6)$$

where $\widehat{\mathbf{M}}(\mathbf{x}_0) = \begin{bmatrix} \mathbf{M}(\mathbf{x}_0) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$, $\widehat{\mathbf{C}}(\mathbf{x}_0) = \begin{bmatrix} \mathbf{C}(\mathbf{x}_0) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$, $\widehat{\mathbf{K}}(\mathbf{x}_0, \boldsymbol{\lambda}_0) = \begin{bmatrix} \mathbf{K}(\mathbf{x}_0, \boldsymbol{\lambda}_0) & \mathbf{G}(\mathbf{x}_0) \\ \mathbf{G}^T(\mathbf{x}_0) & \mathbf{0} \end{bmatrix}$, $\mathbf{M}(\mathbf{x}_0) = (\partial \mathbf{F} / \partial \ddot{\mathbf{x}})(\mathbf{x}_0)$, $\mathbf{C}(\mathbf{x}_0) = (\partial \mathbf{F} / \partial \dot{\mathbf{x}})(\mathbf{x}_0)$, $\mathbf{K}(\mathbf{x}_0, \boldsymbol{\lambda}_0) = (\partial(\mathbf{F} + \mathbf{G}\boldsymbol{\lambda}) / \partial \mathbf{x})(\mathbf{x}_0, \boldsymbol{\lambda}_0)$, and $\delta \mathbf{y} = [\delta \mathbf{x} \ \delta \boldsymbol{\lambda}]^T$.

Writing the solution as $\delta \mathbf{y} = e^{rt} \mathbf{V}$, where r is one of the eigenvalues, $\mathbf{V}$ is the corresponding eigenvector and the eigenvalue problem becomes

$$(r^2 \widehat{\mathbf{M}} + r \widehat{\mathbf{C}} + \widehat{\mathbf{K}}) \mathbf{V} = 0. \quad (7)$$

This equation represents a standard eigenvalue problem that could be solved using standard solver. The frequency shift could be used to insure that $\mathbf{K}$ is nonsingular.

Avilés et al. [21] presented a procedure for the solution of the nonlinear static equilibrium problem in complex multibody mechanical systems, including rigid and elastic elements. The error function was simple based on the potential function and set of nonlinear constraints. Lagrange multipliers along with various versions of the augmented Lagrange multipliers were used to form the system dynamic equations. A Newton-Raphson second-order method is used for seeking function minima for equilibrium positions. Wang et al. [22] used the bond graph to improve the efficiency for the kintostatic analysis of complex multibody systems. They proposed an effective decoupling method to simplify the resulting complexity of the equation of motion. As per Giampiero and Manfred [1], the linearized equation of motion will lead to nonlinear equations that can be solved using Newton's method.

Similarly, commercial multibody dynamic simulation solvers like SIMPACK use Newton-Raphson based solver for nonlinear equations and find the system equilibrium positions and/or calculate the preload in the springs and bushing elements. This solution approach has been used for trains, automotive, and airplane systems. Another approach to determine the static equilibrium is the quasistatic approach. In multibody systems, this approach may not be very efficient if there is significant difference in the inertia of the bodies in the system or if there are dynamic oscillators in the system.

This paper describes a general purpose formulation and implementation for modeling rigid and flexible body in multibody system based on the joint coordinates formulation. A new approach for solving static equilibrium will be presented in this paper. The presented static solver utilizes a recursive transient solver to evaluate the system equations of motion. An energy dissipative damping is introduced to eliminate the system oscillations around the equilibrium position. The system states are dampened using weighted damping factor that is function of the state velocity and the state error. The attributes associated with this method are the ability to handle large heterogeneous systems and ability to linearize the system in terms of arbitrary user-defined coordinates in a straightforward implementation. The spatial algebra operators are used to formulate the kinematic and dynamic equations of motion.

This paper is organized as follows. In the following section, the structure of the equation of motion of the multibody system using joint coordinate formulation is introduced for rigid body system based on the spatial algebra operator. In Section 3, the flexible body kinematic and dynamic equations of motion are presented. In Section 4, the equations of motion of the constrained multibody systems with closed loops will be presented. Section 5 outlines the constraint

enforcement technique based on Baumgarte stabilization approach. Section 6 presents the proposed static equilibrium technique based on the energy drainage. Section 7 outlines the recursive algorithm implemented to solve the proposed multibody system equations of motion. Section 8 presents sample results of the proposed static solution approach for multibody systems. Finally, this paper is summarized and some conclusions are drawn in Section 9.

2. Multibody Formulation and System Equation of Motion

Figure 1 shows an illustration of the typical structure of articulated earth-moving equipment, a wheel loader. The major dynamic components of the machine, as shown in the figure, include the rear frame which contains the engine and drive train, the right and left rear wheels which are connected to the rear frame, the front frame which articulates with respect to the rear frame to steer the machine, the boom which is connected to the front frame and is driven by hydraulic cylinders, the bucket which is connected to the boom and is controlled by other hydraulic cylinders, and the left and right front wheels which are connected to the front frames. This machine model will be used to relate the theoretical multibody equations to physical model for clarification.

The proposed formulation in this investigation utilizes a joint coordinates based multibody approach. The kinematic tree or connectivity graph is developed to describe the system topology based on the connectivity between the different bodies in the system. The ground body (the inertial body) is considered the root (base) of the kinematic tree. Any dynamic body in the system is connected or referenced to one parent body through an arc joint that allows one or more DoF. While each body can have only one parent (ancestor body), it could have one or more child bodies (descendant bodies). In the kinematic tree the root body is numbered as 0 while the descendant bodies are numbered consecutively from 1 to n_b . The body and the joint connecting it to its parent are given the same number [10, 23]. It should be mentioned that the kinematic tree is not unique. Using the parent-child relation, a parent-child list could be developed and stored to be used later in the recursive calculations [13, 24, 25]. Any arbitrary body in the system could be modeled as rigid or flexible body. Figure 2 shows the kinematic tree of the wheel loader under investigation. The dynamic bodies are represented by the circles and the joints are represented by solid arrows. The rear frame is considered the base of our kinematic tree and is referenced to the ground with free joint (joint with 6-DoF). The rear wheels and the front frame are considered descendants of the rear frame and connected to their parent by revolute joints. The remaining bodies in the physical model are shown in the kinematic tree. It should be mentioned that, although the bucket and the boom are driven by two hydraulic cylinders each, only one cylinder is shown in the kinematic tree for simplicity,

$$\text{parent} = \text{list}(k) = \{0, 1, 1, 1, 2, 2, 2, 2, 2, 7, 8, 9\}. \quad (8)$$

FIGURE 1: Typical structure of a wheel loader.

FIGURE 2: Kinematic tree of the wheel loader model.

In the proposed formulation the child body is joined to its parent through two markers; as shown in Figure 3, one marker is attached to the parent side called connector while the other marker is on the child side called the output marker. The output marker is the reference frame of the child body. The body kinematic quantities and dynamic matrices are expressed with respect to the body reference

frame. The position and orientation of the connectors are measured relative to the parent reference marker through the spatial transformation matrix, as shown in Figure 3. The joint degrees of freedom, representing the joint variables, are defined as the allowed relative motion between the two connecting markers. Relative motion between the markers could be translation or rotation along one of the connector marker axes. Massless markers could be inserted between the two bodies to represent joints with more than one DoF. The joint variables displacement, velocities, and accelerations are used as the body states. The Cartesian displacements, velocities, acceleration vectors, and the joint reaction forces are considered as augmented algebraic variables.

The position vector of body i is defined in the global coordinate system by position vector of the origin of the body reference marker $\underline{r}_{0i,i}^0$, as shown in Figure 3, while the orientation of the body could be described by the 3×3 rotation matrix $\underline{R}^{0i}$ which is a function of set of spatial rotational angles. The spatial velocity vector of body i , $\underline{v}_{0i,i}^0 = [\underline{\omega}_{0i}^0 \quad \underline{\dot{r}}_{0i,i}^0]^T$, could be obtained by differentiating the position vector and the body orientation parameters. As the recursive formulation is based on relative motion, the relative spatial velocity vector of the reference frame of a child body j with respect to its parent rigid body i observed at the origin of the local coordinate system of body j can be written as

$$\underline{v}_{ij,j}^j = [\underline{\omega}_{ij}^j \quad \underline{\dot{r}}_{ij,j}^j]^T, \quad (9)$$

FIGURE 3: Body kinematics and joint connections.

where $\dot{\omega}_{ij}^j$ is the time derivative of the orientation parameters of body j with respect to body i and $\dot{r}_{ij,j}^j$ is the derivative of the position vector of body j with respect to body i . The relative velocity across the joint between body i and body j can be written as follows:

$$\mathbf{v}_{ij,j}^j = \mathbf{h}^j \dot{\mathbf{q}}_j, \quad (10)$$

where $\mathbf{h}^j$ is a 6 by n_d joint influence coefficient matrix or partial velocity matrix corresponding to particular columns of the identity matrix depending on which n_d primitive degrees of freedom are represented by $\mathbf{q}_j$, as shown in Figure 3, and $\mathbf{q}_j$ and $\dot{\mathbf{q}}_j$ are the vector of joint variables and its time derivatives.

The spatial velocity vector can be transformed from the CS at j into the CS at i using spatial transformation matrix as follows [9, 10, 24]:

$$\mathbf{v}^i = X^{ij} \mathbf{v}^j, \quad (11)$$

where X^{ij} is the spatial transformation matrix and i and j indicate the coordinate systems. The general transformation matrix X^{ij} is the result of spatial rotation followed by a spatial translation as follows:

$$X^{ij} = L^{ij} R^{ij} = \begin{bmatrix} I & 0 \\ \tilde{r}_{ij}^i & I \end{bmatrix} \begin{bmatrix} R^{ij} & 0 \\ 0 & R^{ij} \end{bmatrix} = \begin{bmatrix} \underline{R}^{ij} & 0 \\ \underline{R}^{ij} \tilde{r}_{ij}^i & \underline{R}^{ij} \end{bmatrix}, \quad (12)$$

where L^{ij} is the 6×6 spatial translation matrix, $\tilde{r}_{ij}^i$ is the 3×1 translational displacement vector of CS j relative to CS i defined in the CS i , $\tilde{r}_{ij}^i$ is 3×3 skew symmetric matrix representing the cross-product operation of $\tilde{r}_{ij}^i$, R^{ij} is the 6×6 spatial rotation matrix, and $\underline{R}^{ij}$ is the 3×3 transformation matrix relating the coordinate frames i and j .

The spatial velocity vector of body j defined in the global coordinate system could be computed as follows:

$$\mathbf{v}_{0,j,0}^0 = X^{0j} \mathbf{v}_{0,j,j}^j, \quad (13)$$

$$\text{where } \mathbf{v}_{0,j,j}^j = X^{ji} \mathbf{v}_{0i,i}^i + \mathbf{v}_{ij,j}^j = \mathbf{v}_{0i,j}^j + \mathbf{h}_{j,j}^j \dot{\mathbf{q}}_j.$$

The spatial velocity vector of a descendant body could be calculated recursively using its parent spatial velocity as follows:

$$\mathbf{v}_{0,j,0}^0 = \mathbf{v}_{0i,0}^0 + \mathbf{v}_{ij,0}^0 = \mathbf{v}_{0i,0}^0 + X^{0j} \mathbf{h}^j \dot{\mathbf{q}}_j. \quad (14)$$

Assuming the system shown in Figure 3 is modeled using rigid bodies, the velocity of body l could be written in a similar way as follows:

$$\mathbf{v}_{0l,l}^l = X^{lj} \mathbf{v}_{0j,j}^j + \mathbf{h}_{j,l}^l \dot{\mathbf{q}}_j. \quad (15)$$

Rearranging the terms in the velocity equations, the recursive form of the system velocity could be expressed as follows [13]:

$$\begin{bmatrix} I & 0 & 0 \\ -X^{ji} & I & 0 \\ 0 & -X^{lj} & I \end{bmatrix} \begin{bmatrix} \mathbf{v}_{0i,i}^i \\ \mathbf{v}_{0j,j}^j \\ \mathbf{v}_{0l,l}^l \end{bmatrix} = \begin{bmatrix} \mathbf{h}^i & 0 & 0 \\ 0 & \mathbf{h}^j & 0 \\ 0 & 0 & \mathbf{h}^l \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_i \\ \dot{\mathbf{q}}_j \\ \dot{\mathbf{q}}_l \end{bmatrix}. \quad (16)$$

The velocity recursive equations in (16) could be written in a compact form as follows:

$$T_a^{\ell\ell} \mathbf{v}_a^\ell = \mathbf{H}_a^\ell \dot{\mathbf{q}}_a = \mathbf{v}_r^\ell, \quad (17)$$

where $T_a^{\ell\ell}$ represents the assembled system topology or connectivity matrix which could be constructed from the connectivity graph, the superscript ℓ refers to local matrices and the superscript a refers to the system assembly matrix, and $\mathbf{H}_a^\ell$ is the assembly influence coefficient matrices grouped in blocks corresponding to the joints' DoF.

It should be mentioned that $T_a^{\ell\ell}$ is a lower triangular matrix and thus has simple inverse that maintains the topological structure of the original matrix and becomes an upper triangular matrix as follows [26]:

$$(T_a^{\ell\ell})^{-1} = \begin{bmatrix} I & -(X^{ji})^T & 0 \\ 0 & I & -(X^{lj})^T \\ 0 & 0 & I \end{bmatrix}. \quad (18)$$

Similarly the spatial acceleration vector of body j could be obtained by differentiating the velocity vector defined in (14) as follows:

$$\mathbf{a}_{0j,0}^0 = \mathbf{a}_{0i,0}^0 + \mathbf{a}_{ij,0}^0, \quad (19)$$

$$\text{where, } \mathbf{a}_{0j,0}^0 = \mathbf{h}_{j,0}^0 \ddot{\mathbf{q}}_j + \dot{\mathbf{v}}_{0j,0}^0 \mathbf{h}_{j,0}^0 \dot{\mathbf{q}}_j.$$

Assuming that the quadratic velocity term could be expressed as $\gamma_{0j,0}^0 = \dot{\mathbf{v}}_{0j,0}^0 \mathbf{h}_{j,0}^0 \dot{\mathbf{q}}_j = \dot{\mathbf{v}}_{0j,0}^0 \mathbf{v}_{0j,0}^0$, the global acceleration vector could be written as follows:

$$\mathbf{a}_{0j,0}^0 = \mathbf{a}_{0i,0}^0 + \mathbf{a}_{ij,0}^0 = \mathbf{a}_{0i,0}^0 + \mathbf{h}_{j,0}^0 \ddot{\mathbf{q}}_j + \gamma_{0j,0}^0. \quad (20)$$

The recursive form to calculate the system local accelerations could be written as follows:

$$T_a^{\ell\ell} \mathbf{a}_a^\ell = \mathbf{H}_a^\ell \ddot{\mathbf{q}}_a + \dot{\mathbf{H}}_a^\ell \dot{\mathbf{q}}_a = \mathbf{H}_a^\ell \ddot{\mathbf{q}}_a + \tilde{\mathbf{v}}_r^\ell \mathbf{v}_r^\ell = \mathbf{H}_a^\ell \ddot{\mathbf{q}}_a + \gamma_a^\ell. \quad (21)$$

Equations (17) and (21) could be used to recursively calculate the velocity and acceleration of the system bodies marching outward from the root to the descendant bodies. Similarly, the system topological matrix could be used to project the joint forces in the joint subspace as follows:

$$\mathbf{Q}_a = (\mathbf{H}_a^l)^T \mathbf{F}_a^l. \quad (22)$$

The kinematic equations derived earlier could be used to express the rigid body equations of motion from the momentum equations. The spatial momentum vector of the rigid body j can be written as follows:

$$\mathbf{P}_{j,G}^j = M_{j,GG}^{jj} \mathbf{v}_{0j,G}^j, \quad (23)$$

where $\mathbf{P}_{j,G}^j$ is the spatial momentum of body j , $M_{j,GG}^{jj}$ is the spatial mass matrix defined at the body center of mass marker G , and $\mathbf{v}_{0j,G}^j$ is the global velocity of the marker G located at the body center of mass. The centroidal spatial mass matrix is defined as follows:

$$M_{j,GG}^{jj} = \begin{bmatrix} J_{j,GG}^{jj} & 0 \\ 0 & m_j \mathbf{I} \end{bmatrix}, \quad (24)$$

where m_j is the mass body j , $\mathbf{I}$ is 3×3 identity matrix, and $J_{j,GG}^{jj}$ is the 3×3 matrix representing the body j moment of inertia tensor defined at a marker located at the body center of mass, G . This rigid body momentum can be transformed into the global coordinate system and differentiated with respect to time to obtain the body equation of motion as follows:

$$M_{j,00}^{00} \mathbf{a}_{0j,0}^0 - \dot{\mathbf{v}}_{0j,0}^{0T} M_{j,00}^{00} \mathbf{v}_{0j,0}^0 = \mathbf{G}_{j,0}^0, \quad (25)$$

where the symbol 0 refers to global coordinate system, $\mathbf{G}_{j,0}^0 = [\mathbf{r}_{j,0}^0 \ \underline{g}_j^0]^T$ is the spatial force vector including the gravitational forces, $\underline{g}_j^0$ is the sum of all external forces acting on the body j , and $\mathbf{r}_{j,0}^0$ contains the sum of all torques and the moments of all forces about the origin of frame 0.

The equation of motion of any two connected bodies i and j could be driven from the free body diagram as follows. For body j , the equation of motion in its local coordinate system could be written as

$$M_{j,jj}^{jj} \mathbf{a}_{0j,j}^j - \mathbf{F}_{j,j}^j = \mathbf{G}_{j,j}^j + \tilde{\mathbf{v}}_{0j,j}^{jT} \mathbf{P}_{j,j}^j, \quad (26)$$

where $\mathbf{F}_{j,j}^j$ is the vector of reaction forces of the joint connecting body j to its parent. While for body i , the equation of motion in its local coordinate system could be written as

$$M_{i,ii}^{ii} \mathbf{a}_{0i,i}^i - \mathbf{F}_{i,i}^i + (X^{ji})^T \mathbf{F}_{j,j}^j = \mathbf{G}_{i,i}^i + (\tilde{\mathbf{v}}_{0i,i}^i)^T \mathbf{P}_{i,i}^i. \quad (27)$$

The equations could be rearranged in matrix form as follows:

$$\begin{bmatrix} M_{i,ii}^{ii} & 0 \\ 0 & M_{j,jj}^{jj} \end{bmatrix} \begin{bmatrix} \mathbf{a}_{0i,i}^i \\ \mathbf{a}_{0j,j}^j \end{bmatrix} + \begin{bmatrix} I & -(X^{ji})^T \\ 0 & I \end{bmatrix} \begin{bmatrix} -\mathbf{F}_{i,i}^i \\ -\mathbf{F}_{j,j}^j \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{i,i}^i + (\tilde{\mathbf{v}}_{0i,i}^i)^T \mathbf{P}_{i,i}^i \\ \mathbf{G}_{j,j}^j + (\tilde{\mathbf{v}}_{0j,j}^j)^T \mathbf{P}_{j,j}^j \end{bmatrix}. \quad (28)$$

Similar to the velocity and acceleration equations (14), (15), (16), and (17), it could be shown that the connectivity matrix could be used to simplify the expression of the system equations of motion. The system equation of motion could be written for the whole system assembly in a more compact form as follows:

$$M^{\ell\ell} \mathbf{a}_a^\ell - (T_a^{\ell\ell})^T \mathbf{F}_a^\ell = \mathbf{G}_a^\ell + (\tilde{\mathbf{v}}_a^\ell)^T \mathbf{P}_a^\ell. \quad (29)$$

The system of equations in (29) contains the unknown Cartesian acceleration of the bodies as well as the joint reaction forces. In order to be able to solve the equation of motion, the kinematic relations from (21), the constraint equations from (22), and the system dynamic equation (29) are rearranged in matrix form as follows:

$$\begin{bmatrix} M^{\ell\ell} & 0 & (T_a^{\ell\ell})^T \\ 0 & 0 & -\mathbf{H}_a^\ell \\ T_a^{\ell\ell} & -(\mathbf{H}_a^\ell)^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a}_a^\ell \\ \ddot{\mathbf{q}}_a^\ell \\ -\mathbf{F}_a^\ell \end{bmatrix} = \begin{bmatrix} \mathbf{G}_a^\ell + (\tilde{\mathbf{v}}_a^\ell)^T \mathbf{P}_a^\ell \\ \mathbf{Q}_a^\ell \\ \gamma_a^\ell \end{bmatrix}, \quad (30)$$

where $M^{\ell\ell}$ is a block diagonal inertia matrix expressed in the Cartesian form, $T_a^{\ell\ell}$ is the body connectivity matrix, $\mathbf{H}_a^\ell$ is a block diagonal joint influence coefficient matrix, $\mathbf{a}_a^\ell$ is the body Cartesian accelerations, $\mathbf{F}_a^\ell$ is a vector of the Lagrange multipliers, $\ddot{\mathbf{q}}_a^\ell$ is the second time derivative of joint variables, $\mathbf{G}_a^\ell$ is a vector of the externally applied loads, γ_a^ℓ is the quadratic constraint derivatives, and $\mathbf{Q}_a^\ell$ is the generalized

forces. Since the inertia matrix M^{el} is invertible, (30) could be solved to get the unknown joint accelerations $\ddot{\mathbf{q}}_a$ which can be used to calculate the joint Cartesian accelerations, $\mathbf{a}_a^e$, and the joint reaction forces $\mathbf{F}_a^e$ as follows:

$$\begin{aligned} & \left((\mathbf{H}_a^e)^T (\mathbf{T}_a^{el})^{-T} \mathbf{M}^{el} (\mathbf{T}_a^{el})^{-1} \mathbf{H}_a^e \right) \ddot{\mathbf{q}}_a \\ & = \mathbf{Q}_a + (\mathbf{H}_a^e)^T (\mathbf{T}_a^{el})^{-T} \left(\mathbf{G}_a^e + (\ddot{\mathbf{v}}_a^e)^T \mathbf{P}_a^e \right). \end{aligned} \quad (31)$$

Equation (31) could be written in a compact form as follows:

$$\mathbf{M}_{qq} \ddot{\mathbf{q}}_a = \mathbf{Q}, \quad (32)$$

where $\mathbf{M}_{qq}$ is the generalized mass matrix projected in the joint subspace and can be calculated as follows:

$$\mathbf{M}_{qq} = (\mathbf{H}_a^e)^T (\mathbf{T}_a^{el})^{-T} \mathbf{M}^{el} (\mathbf{T}_a^{el})^{-1} \mathbf{H}_a^e \quad (33)$$

and $\mathbf{Q}$ is the generalized force vector projected in the joint subspace and can be calculated as follows:

$$\mathbf{Q} = \mathbf{Q}_a + (\mathbf{H}_a^e)^T (\mathbf{T}_a^{el})^{-T} \left(\mathbf{G}_a^e + (\ddot{\mathbf{v}}_a^e)^T \mathbf{P}_a^e \right). \quad (34)$$

This equation of motion (32) represents minimum set of differential equations that are function of the system generalized coordinates. However, this form is limited to open-loop unconstrained mechanisms and therefore is suitable only for robotic type mechanisms.

3. Equation of Motion of Rigid and Flexible Body System

The flexible body dynamics can be modeled in the multibody system as a reduced form of the finite element model. The finite element can be reduced using nodal approach or modal approach. Modal formulation uses a set of kinematically admissible modes to represent the deformation of the flexible body [27, 28]. Component mode synthesis approach is widely used in the multibody dynamics codes to accurately simulate the flexible body dynamics in the multibody system. The main advantage of the modal formulation is that fewer elastic modes can be used to accurately capture the flexible body dynamics at reasonable computational cost [29, 30]. Craig-Bampton approach was introduced to account for the effect of boundary conditions and the attachment joints of the flexible body [31, 32]. The structure of the reduced flexible body equation-of-motion can be written as follows [23]:

$$\begin{bmatrix} M_{rb}^{rr} & M^{rf} \\ M^{fr} & M^{ff} \end{bmatrix} \begin{bmatrix} \mathbf{a}^r \\ \ddot{\mathbf{q}}^f \end{bmatrix} = \begin{bmatrix} \mathbf{G}^r \\ \mathbf{Q}^f \end{bmatrix}, \quad (35)$$

where r refers to the flexible body's local reference frame, f refers to flexible or elastic coordinates, M_{rb}^{rr} represents the 6×6 inertia matrix associated with the reference motion, M^{rf} , M^{fr} are the inertia coupling terms between the reference motion and the elastic coordinates, M^{ff} is the inertia matrix associated with the elastic coordinates, $\mathbf{a}^r$ is the reference

frame acceleration in the Cartesian space, $\mathbf{q}^f$ represents the elastic displacements measured relative to that frame, $\mathbf{G}^r$ is the vector of external forces including centrifugal and Coriolis forces [1, 23, 27], and $\mathbf{Q}^f$ is the vector of elastic forces projected into the modal space. It should be mentioned that the matrices M_{rb}^{rr} , M^{rf} , and M^{fr} depend on the elastic coordinates and should be updated every time step. To improve the computation efficiency of the flexible body simulation, a set of inertia shape invariants (ISI) are identified and calculated before the simulation starts. The ISI are used during the dynamic simulation to update the inertia terms. The ISI depend on the kinematic expression of the system equations.

A general form of the equation of motion of multibody system with flexible bodies could be written in a compact form as follows [23]:

$$\begin{bmatrix} \mathbf{M}^{rr} & \mathbf{T}^T & \mathbf{M}^{rf} \\ \mathbf{T} & 0 & -\mathbf{H} \\ (\mathbf{M}^{rf})^T & -\mathbf{H}^T & \mathbf{M}^{ff} \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ -\mathbf{F} \\ \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{G} \\ \boldsymbol{\gamma} \\ \mathbf{Q} \end{bmatrix}, \quad (36)$$

where the block diagonal matrix $\mathbf{M}^{rr}$ is composed of six by six inertia matrices that represent the Cartesian inertia matrices associated with the reference frame for the rigid and flexible bodies in the system, $\mathbf{M}^{rf}$ is the inertia coupling terms between the modal elastic coordinates and Cartesian reference accelerations, $\mathbf{M}^{ff}$ is a block diagonal matrix containing the flexible body inertias, $\mathbf{T}$ is a lower triangular topology matrix, $\mathbf{H}$ is a block matrix of the joint influence coefficient matrices, $\mathbf{a}$ is the vector of Cartesian accelerations, $\mathbf{F}$ is the vector of joint reaction forces, $\ddot{\mathbf{q}}$ is the vector of joint accelerations with the appended elastic coordinates, $\mathbf{G}$ is the vector of external forces, $\boldsymbol{\gamma}$ is a column matrix of 6×1 spatial quadratic acceleration vectors, and $\mathbf{Q}$ is the vector of joint and elastic forces.

The system of equations given by (36) can be easily solved for the joint accelerations and the modal coordinates accelerations. Then the Cartesian accelerations and the reaction forces could be easily computed. Since the inverse of system topology matrix $\mathbf{T}$ is just its transpose, manipulating the terms in (36), we can get the joint and modal accelerations as follows:

$$\begin{aligned} & (\mathbf{M}^{ff} + \mathbf{H}^T (\mathbf{T}^T \mathbf{M}^{rr} \mathbf{T}) \mathbf{H} + \mathbf{H}^T \mathbf{T}^T \mathbf{M}^{rf} + \mathbf{M}^{fr} \mathbf{T} \mathbf{H}) \ddot{\mathbf{q}} \\ & = \mathbf{Q} + \mathbf{H}^T \mathbf{T}^T \mathbf{G} - \mathbf{H}^T (\mathbf{T}^T \mathbf{M}^{rr} \mathbf{T}) \boldsymbol{\gamma} - \mathbf{M}^{fr} \mathbf{T} \boldsymbol{\gamma}. \end{aligned} \quad (37)$$

This equation could be written in a compact form as follows:

$$\mathbf{M}_{qq} \ddot{\mathbf{q}}_a = \mathbf{Q}. \quad (38)$$

The Cartesian accelerations could be calculated as follows:

$$\mathbf{a} = \mathbf{T} \mathbf{H} \ddot{\mathbf{q}} + \mathbf{T} \boldsymbol{\gamma}, \quad (39)$$

and the Cartesian joint reaction forces could be obtained from

$$\mathbf{F} = \mathbf{T}^T \mathbf{M}^{rf} + (\mathbf{T}^T \mathbf{M}^{aa} \mathbf{T}) \mathbf{H} \ddot{\mathbf{q}} - \mathbf{T}^T \mathbf{G} + (\mathbf{T}^T \mathbf{M}^{aa} \mathbf{T}) \boldsymbol{\gamma}. \quad (40)$$

The sequence of evaluating the terms in (37) to (40) could be optimized in order to minimize the computational efforts and to avoid repeated calculations. Both (32) and (38) represent the minimum set of differential equations that can completely describe the system dynamics.

4. Equation of Motion of Constrained System

The abovementioned development could be efficiently used to model open-loop systems like robots and human body systems. In the open-loop systems, each body is connected with one joint only. This open-loop systems approach will not be adequate to model the wheel loader system shown in Figure 1. In the kinematic tree, shown in Figure 2, the solid arrows represent a joint where the dotted lines represent a joint that will form a closed kinematic loop. According to this description, the kinematic tree of the wheel loader contains two closed loops formed of the following list of bodies: $\Gamma^1 = \{2, 8, 11, 9\}$ and $\Gamma^2 = \{2, 9, 12, 10, 7\}$. Each of those two loops could be represented by a set of constraint equations that are function of the joint variable of all the bodies forming the loop as follows: $\Phi^1(\mathbf{q}_2, \mathbf{q}_8, \mathbf{q}_{11}, \mathbf{q}_9) = 0$ and $\Phi^2(\mathbf{q}_2, \mathbf{q}_9, \mathbf{q}_{12}, \mathbf{q}_{10}, \mathbf{q}_7) = 0$. The first and second derivatives of the constraint equations could be easily evaluated from the kinematic equations. The multibody system's equations of motion with appended constraints could be formed as follows:

$$\begin{bmatrix} \mathbf{M}_{qq} & \mathbf{J}^T \\ \mathbf{J} & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_a \\ \lambda_q \end{bmatrix} = \begin{bmatrix} \mathbf{Q} \\ \gamma_q \end{bmatrix}, \quad (41)$$

where $\mathbf{M}_{qq}$ is the generalized mass matrix from (32) and (38), $\mathbf{J}$ is the Jacobian matrix, and the other column matrices are similar to those described earlier.

The coefficient matrix in the left hand side of (41) is often singular, and general-purpose sparse matrix solver algorithms are routinely used to analyze, factor, and solve the equations. An alternative approach based on generalized coordinate partitioning [14] may be employed at this stage. The factored forms of (33) and (34) used to obtain (32) play a significant role in the solver efficiency. The generalized coordinate partitioning method [14] first analyzes the model topology and decomposes it into a number of kinematically uncoupled subelements, each with a small Jacobian matrix [2]. The uncoupled acceleration constraint equations could be arranged as follows:

$$\begin{bmatrix} \Phi_q^1 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & \Phi_q^n \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}^1 \\ \vdots \\ \ddot{\mathbf{q}}^n \end{bmatrix} = \begin{bmatrix} \gamma^1 \\ \vdots \\ \gamma^n \end{bmatrix}. \quad (42)$$

The Jacobian matrices in (42) are factored into LU form using complete pivoting to partition the variables into dependent and independent sets. The results are then combined into a revised set of constraint equations [33]

$$\ddot{\mathbf{q}}^d = \mathbf{B}\ddot{\mathbf{q}}^i + \gamma^d, \quad (43)$$

where $\ddot{\mathbf{q}}^d$ is the vector of dependent variables and $\ddot{\mathbf{q}}^i$ is the vector of independent variables. Appending (43) with a set

of Lagrange multipliers to the partitioned and permuted equations of motion gives

$$\begin{bmatrix} \mathbf{M}_q^{dd} & \mathbf{I} & \mathbf{M}_q^{di} \\ \mathbf{I} & 0 & -\mathbf{B} \\ \mathbf{M}_q^{id} & -\mathbf{B}^T & \mathbf{M}_q^{ii} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}^d \\ \lambda^d \\ \ddot{\mathbf{q}}^i \end{bmatrix} = \begin{bmatrix} \mathbf{Q}^d \\ \gamma^d \\ \mathbf{Q}^i \end{bmatrix}. \quad (44)$$

The coefficient matrix in the upper-left part of (44) is structurally nonsingular, so it can be reduced to

$$\overline{\mathbf{M}}_q^{ii} \ddot{\mathbf{q}}^i = \overline{\mathbf{Q}}^i, \quad (45)$$

where

$$\begin{aligned} \overline{\mathbf{M}}_q^{ii} &= \mathbf{M}_q^{ii} + \mathbf{M}_q^{id} \mathbf{B} + \mathbf{B}^T \mathbf{M}_q^{di} + \mathbf{B}^T \mathbf{M}_q^{dd} \mathbf{B}, \\ \overline{\mathbf{Q}}^i &= (\mathbf{Q}^i - \mathbf{M}_q^{id} \gamma^d) + \mathbf{B}^T (\mathbf{Q}^d - \mathbf{M}_q^{dd} \gamma^d). \end{aligned} \quad (46)$$

Equation (45) could be solved by Cholesky factorization.

5. Constraint Enforcement

The equations of motion in (41) and (44) are index-2 differential algebraic equations (DAE) [1]. This set of DAE are traditionally linearized and solved by implicit numerical integrators [34–36]. The generalized coordinate partitioning approach [14] is used to avoid the complexity and cost of implicit solvers, but this approach does not guarantee continued satisfaction of the system constraints. Baumgarte [37] introduced a constraint stabilization method based on the control theory. In this approach, the original acceleration level constraints are replaced with

$$\ddot{\Phi} + \alpha \dot{\Phi} + \beta \Phi = 0, \quad (47)$$

where α and β are often chosen to appropriately achieve critical damping. The Baumgarte stabilization is implemented in the following form:

$$\ddot{\Phi} + 2\omega_n \dot{\Phi} + \omega_n^2 \Phi = 0, \quad (48)$$

where ω_n is a user-specified gain (natural frequency). Unfortunately, the value of the parameter ω_n is problem dependent, and no general procedure exists for its determination. If $\Phi = \dot{\Phi} = \ddot{\Phi} = 0$ represents the constraint equations associated with (41), then substituting (48) into (41) leads to

$$\begin{bmatrix} \mathbf{M}_q & \mathbf{J}^T \\ \mathbf{J} & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda_q \end{bmatrix} = \begin{bmatrix} \mathbf{Q} \\ \gamma_q - 2\omega_n \dot{\Phi} - \omega_n^2 \Phi \end{bmatrix}. \quad (49)$$

A similar modification would be made to the right hand side of (44). The value set for ω_n is critical and affects the integrator performance. If ω_n is too large, constraint violations will be very small, but the equation of motion will tend to be stiff and the integrator will be slow. If ω_n is too small, the stabilization algorithm will not efficiently reduce the constraint violations. From experience, the maximum value of best ω_n is probably near the highest system frequencies in the system. To this end, the developed set of dynamic equations of motion is sufficient to perform the dynamic simulation of open- and closed-loop systems. It is also sufficient for modeling rigid and flexible bodies in the multibody system.

6. Static Equilibrium Analysis

The mechanical system can be considered at static equilibrium if the kinetic energy is zero or constant target value and the potential energy is minimum. Many methods have been proposed to solve static equilibrium in nonlinear multibody system models. Most of those proposed techniques require minimizing potential functions and/or linearizing the equations of motion [36] as mentioned earlier. Writing potential functions for arbitrary nonlinear force modules or linearizing them is a daunting task. The problem becomes more challenging when models contain friction and other history-based force models because unique iterative static solutions may not exist. The proposed approach uses the dynamic model itself to solve the static equilibrium position. This approach may have some drawbacks because convergence can be slow in lightly damped systems. In order to improve the solver efficiency, an energy-drainage mechanism is used.

In the energy-drainage approach, a technique similar to the Baumgarte equation was implemented into the solver. After solving (45) to obtain the accelerations $\ddot{\mathbf{q}}$, the solver imposes the additional requirement on the second derivatives

$$\ddot{q}_d = -2\omega_d \dot{q}_i - \omega_d^2 (q_i - q_{i-1}), \quad (50)$$

where ω_d is a natural damping frequency and q_i and q_{i-1} represent the system states at times t_i and t_{i-1} . Choosing ω_d is tricky as it was in the constraint stabilization. If ω_d is too small, the damping will be weak and the model will oscillate too long. And if ω_d is too large, damping will be strong and the model will drift slowly toward equilibrium. With this damping method, equilibrium is approached as $\|q_i - q_{i-1}\|$ gets acceptably small, not necessarily when the kinetic energy gets small.

Another feature of energy-drainage damping is its ability to inhibit the effects of bad initial conditions that often cause models to fly apart and the numerical integrator to fail. Extremely high forces from the offending contact models can cause unrealistic accelerations, which are then integrated to give unrealistic velocities and displacements. When energy-drainage damping is active, the resulting large system velocities are immediately fed back through the above damping equation, and this generates large opposing acceleration terms, canceling those from the model. With energy-drainage damping, the contacting bodies will tend to drift slowly apart until the forces and accelerations become reasonable.

7. Recursive Algorithm

At the beginning of the dynamic simulation, the structure of the equation of motion is determined based on a preliminary analysis of the system topology. The dependent and independent variable sets are determined using the generalized coordinate portioning approach. The solver integrates both dependent and independent variable sets and the kinematic constraints are enforced using Baumgarte stabilization, as explained in the previous section. The input states to the integrator represent the first and second time derivative of

the joint variables and the output states are the joint displacements and the joint velocities. The recursive algorithm of the multibody dynamic simulation can be summarized as follows.

- (1) Using the joint variables which returned from the integrator, the solver calculates the Cartesian displacements, velocities, and accelerations. Forward evaluation scheme is utilized (starting from the root body to the descendant/branch bodies).
- (2) Update and factor the Jacobian matrix.
- (3) Apply generalized coordinate portioning technique to determine the quality of the independent and dependent set of variables.
- (4) Calculate the constraint violations and apply the Baumgarte stabilization.
- (5) Calculate the internal and external forces and apply them to the different bodies (interaction forces between bodies, driver forces, soil/terrain forces, etc.).
- (6) Transform the inertia matrices into global coordinate system.
- (7) Calculate the inertia forces and centrifugal and Coriolis forces in Cartesian space.
- (8) Propagate the external and inertia forces from the descendant bodies into their parents.
- (9) Project the Cartesian forces into the body joint space.
- (10) Factor the mass matrix based on the current independent and dependent variables selection.
- (11) Calculate second derivatives of the joint variables.
- (12) Apply the energy drainage operator according to (50).
- (13) Send the states to the integrator.

The solver utilizes a predictor-corrector integrator with variable-order interpolation polynomials and variable time step. This explicit integrator insures the stability of the solution and the ability to capture the high-speed impacts between the different machine parts.

8. Examples

This section presents some examples of the simulation results multibody system ranging from simple pendulum to a full wheeled machine. The first example represents a uniform beam connected to the ground with a revolute joint to form a single pendulum, as shown in Figure 4. The link has mass of 1.7595 Kg. The beam center of mass is located in the middle. The pendulum was originally horizontal and is allowed to fall under the effect of gravity. The static equilibrium position of this pendulum is known to be vertical.

The proposed multibody formulation and solver were used to model the pendulum and compute the static equilibrium position. The transient simulation is run for a few iterations to allow the system to pick up some kinetic energy. Once the pendulum starts to move the energy draining mechanism is applied. A stopping criterion is set to be a target

FIGURE 4: Single pendulum beam.

FIGURE 5: The pendulum angle as it falls under the effect of gravity.

FIGURE 6: The velocity and kinetic energy of the pendulum.

value of the kinetic energy. When the pendulum reaches this target value, it would have reached the static equilibrium.

Figure 5 shows the value of the pendulum angle as it falls under the effect of gravity till the system reaches equilibrium position. The kinetic energy is drained out from the system as shown in Figure 6.

The change in the pendulum potential energy from the reference is shown in Figure 7. As the pendulum starts moving, the potential energy of the pendulum decreases

FIGURE 7: The potential energy of the pendulum.

FIGURE 8: Kinetic energy of the 50-link chain.

and the minimum value is reached at the static equilibrium position. The final value of the potential energy is -12.9455 J which matches the closed form solution for the pendulum.

It should be mentioned that time in the plots is only a representation of the number of iteration to achieve solution convergence. Also, it should be mentioned that this example represents an extreme case where the pendulum initial position was extremely far from the equilibrium position. This example shows that the solver is stable and can achieve realistic solutions even in extreme cases.

The second example represents a system that might have minor assembly errors or constraint violations. The system represents a chain of 50 links connected with revolute joints. The chain links had same properties of the link in first example. In this example, the first link is attached to the ground and it has initial angle of 5° from the vertical position. Every following link has an initial deviation of 1° relative to its parent. The static equilibrium analysis was performed on the system. The kinetic energy of the system is shown in Figure 8. The potential energy of the system is shown in Figure 9 while the angles of all the links are shown in Figure 10.

FIGURE 11: Grapple skidder forestry machine.

FIGURE 12: Kinematic tree of log dragging forestry equipment.

FIGURE 13: Static equilibrium of wheeled machine (grapple skidder) on a washboard surface.

The system assembly influence coefficient matrix could be written as follows:

$$\mathbf{H}_a^\ell = \begin{bmatrix} \mathbf{h}^1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{h}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{h}^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{h}^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{h}^5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{h}^6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{h}^7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{h}^8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{h}^9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{h}^{10} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{h}^{11} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{h}^{12} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{h}^{13} \end{bmatrix}. \quad (53)$$

When the solver starts, the Cartesian velocities and accelerations are calculated using forward substitution from the root body to the descendant bodies. The Jacobian matrices are updated based on the new values of the spatial displacements and velocities. The external forces acting on the system, the contact forces between the different bodies, and the interaction forces between the system and terrain are calculated and applied to each body's force vector. The system topology matrix is used to project the Cartesian quantities into the joint subspace and define the minimum set of the system variables as shown in (32) to (34). The system Jacobian matrix is then appended to the inertia matrix to form the augmented constrained multibody equation of motion. The solver then utilizes the generalized coordinate partitioning approach to identify the dependent and independent variables by factoring the Jacobian matrix into LU form. The independent variable is identified, as shown in (45). The constraint violation penalty is applied to the system equation of motion using Baumgarte equation, as shown in (48). The entries to the mass matrix are permuted based on the order in the LU factorization order.

To perform the static analysis, at this point, the system accelerations are modified using the energy drainage function as shown in (50), and both the dependent and independent variables are sent to the integrator. The integrated states

are used to calculate the Cartesian position, velocities, and accelerations of all bodies. The Cartesian velocities are used to calculate the kinetic energy of the system and the global position of the bodies is used to calculate the potential energies. If the value of the kinetic energy of the system reached the target, the solver stops and reports the current state as the body equilibrium position.

Figure 13 shows a successful simulation to achieve the static equilibrium of wheeled machine. The machine is dropping from a small height over a washboard surface. As the machine drops, the rear tires fall on the inclined plane of the surface causing the machine to roll down and backward. As the machine contacts the opposite surface of the washboard ditch, the machine comes to static equilibrium.

9. Conclusions

This paper presented an approach for evaluating the static equilibrium position of multibody systems. The multibody system can include rigid and flexible bodies. The proposed approach is implemented in a joint-based multibody dynamics formulation that uses the spatial algebra to derive the equation of motion of the multibody system. The kinematic equation of the closed-loop systems was optimized to employ Baumgarte constraint stabilization approach to eliminate the constraint violation while at the same time avoiding using iterative constraint enforcement schemes. In order to determine the system static equilibrium position, an energy drainage mechanism was introduced to modify the system states before integrating it. The static equilibrium is achieved by running a transient simulation of the dynamics system while the integrated states were dampened with the energy drainage operator.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

References

- [1] M. Giampiero and P. Manfred, *Simulation Algorithms in Vehicle System Dynamics*, CRC Press, 2004.

- [2] M. A. Omar, "An applied approach for large-scale multibody dynamics simulation and machine-terrain interaction," *SAE International Journal of Passenger Cars—Electronic and Electrical Systems*, vol. 1, no. 1, pp. 820–828, 2009.
- [3] J. Y. Wong, "Dynamics of tracked vehicles," *Vehicle System Dynamics*, vol. 28, no. 2-3, pp. 197–219, 1997.
- [4] H. C. Lee, J. H. Choi, and A. A. Shabana, "Spatial dynamics of multibody tracked vehicles part II: contact forces and simulation results," *Vehicle System Dynamics*, vol. 29, no. 2, pp. 113–137, 1998.
- [5] Z.-D. Ma and N. C. Perkins, "A track-wheel-terrain interaction model for dynamic simulation of tracked vehicles," *Vehicle System Dynamics*, vol. 37, no. 6, pp. 401–421, 2002.
- [6] Z.-D. Ma and N. C. Perkins, "An efficient multibody dynamics model for internal combustion engine systems," *Multibody System Dynamics*, vol. 10, no. 4, pp. 363–391, 2003.
- [7] P. Roland, D. Daniel, S. Emilio, and Á. N. Miguel, "Validation of a multibody model for an X-by-wire vehicle prototype through field testing," in *Proceedings of the ECCOMAS Thematic Conference on Multibody Dynamics*, J. C. Samin and P. Fisette, Eds., Brussels, Belgium, July 2011.
- [8] A. A. Shabana, *Dynamics of Multibody Systems*, Cambridge University Press, Cambridge, UK, 3rd edition, 2005.
- [9] P. E. Nikravesh, *An Overview of Several Formulations for Multibody Dynamics*, Product Engineering, 2005.
- [10] R. Featherstone, *Rigid Body Dynamics Algorithms*, Springer, New York, NY, USA, 2008.
- [11] R. Featherstone, "The acceleration vector of a rigid body," *International Journal of Robotics Research*, vol. 20, no. 11, pp. 841–846, 2001.
- [12] R. Featherstone and D. Orin, "Robot dynamics: equations and algorithms," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00)*, pp. 826–834, April 2000.
- [13] R. Featherstone, "Efficient factorization of the joint-space inertia matrix for branched kinematic trees," *International Journal of Robotics Research*, vol. 24, no. 6, pp. 487–500, 2005.
- [14] R. A. Wehage and E. J. Haug, "Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems," *Journal of Mechanical Design*, vol. 134, pp. 247–255, 1982.
- [15] R. A. Wehage and M. J. Belczynski, "High resolution vehicle simulations using precomputer coefficients," in *Proceedings of the Winter Annual Meeting of the American Society of Mechanical Engineers*, vol. 44, pp. 311–325, November 1992.
- [16] G. Rodriguez, A. Jain, and K. Kreutz-Delgado, "Spatial operator algebra for multibody system dynamics," *Journal of the Astronautical Sciences*, vol. 40, no. 1, pp. 27–50, 1992.
- [17] G. Rodriguez, A. Jain, and K. Kreutz-Delgado, "Spatial operator algebra for manipulator modeling and control," *International Journal of Robotics Research*, vol. 10, no. 4, pp. 371–381, 1991.
- [18] E. Eich-Soellner and C. Führer, *Numerical Methods in Multibody Dynamics*, Teubner, Stuttgart, Germany, 1998.
- [19] C. Yang, D. Cao, Z. Zhao, Z. Zhang, and G. Ren, "A direct eigenanalysis of multibody system in equilibrium," *Journal of Applied Mathematics*, vol. 2012, Article ID 638546, 12 pages, 2012.
- [20] J. García de Jalón and E. Bayo, *Kinematic and Dynamic Simulation of Multibody Systems, the Real-Time Challenge*, Springer, New-York, NY, USA, 1994.
- [21] R. Avilés, G. Ajuria, V. Gómez-Garay, and S. Navalpotro, "Comparison among nonlinear optimization methods for the static equilibrium analysis of multibody systems with rigid and elastic elements," *Mechanism and Machine Theory*, vol. 35, no. 8, pp. 1151–1168, 2000.
- [22] Z. S. Wang, Y. Y. Tao, and Q. Y. Wen, "A vector bond graph method of kineto-static analysis for spatial multibody systems," *Applied Mechanics and Materials*, vol. 321–324, pp. 1725–1729, 2013.
- [23] M. A. Omar, "Modeling flexible bodies in multibody systems in joint-coordinates formulation using spatial algebra," *Advances in Mechanical Engineering*, vol. 2014, Article ID 468986, 18 pages, 2014.
- [24] R. W. Wehage, "Automated procedures for robust and efficient solution of over-constrained multibody dynamics," in *Proceeding of ASME International Mechanical Engineering Congress and Exposition (IMECE '12)*, vol. 85259, Huston, Tex, USA, November 2012.
- [25] P. E. Nikravesh, "Construction of the equations of motion for multibody dynamics using point and joint coordinates," in *Computer-Aided Analysis of Rigid and Flexible Mechanical Systems*, vol. 268 of NATO ASI Series E: Applied Sciences, pp. 31–60, Kluwer Academic Publishers, 1994.
- [26] A. Jain, "Multibody graph transformations and analysis—part I: tree topology systems," *Nonlinear Dynamics*, vol. 67, no. 4, pp. 2779–2797, 2012.
- [27] M. A. Omar, *Finite Element Modeling of Leaf Springs in Multibody Systems*, VDM Verlag Dr. Müller Aktiengesellschaft & Co. KG, Saarbrücken, Germany, 2010.
- [28] M. A. Omar, A. A. Shabana, A. Mikkola, W.-Y. I. Loh, and R. Basch, "Multibody system modeling of leaf springs," *Journal of Vibration and Control*, vol. 10, no. 11, pp. 1601–1638, 2004.
- [29] K. J. Bathe, *Finite Element Procedures*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1996.
- [30] T. J. R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Dover, New York, NY, USA, 2000.
- [31] R. R. Craig and M. C. Bampton, "Coupling of substructures for dynamic analyses," *AIAA Journal*, vol. 6, no. 7, pp. 1313–1319, 1968.
- [32] R. R. Craig and C. J. Chang, "On the use of attachment modes in substructure coupling for dynamic analysis," in *Proceedings of the 18th Conference on Structures, Structural Dynamics and Materials*, San Diego, Calif, USA, 1977.
- [33] A. A. Shabana, *Computational Dynamics*, John Wiley & Sons, 3rd edition, 2010.
- [34] U. M. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 1998.
- [35] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical Solutions of Initial-Value Problems in Differential-Algebraic Equations*, Society for Industrial and Applied Mathematics, New York, NY, USA, 1996.
- [36] J. García de Jalón and E. Bayo, *Kinematic and Dynamic Simulation of Multibody Systems, the Real-Time Challenge*, Springer, NewYork, NY, USA, 1994.
- [37] J. Baumgarte, "Stabilization of constraints and integrals of motion in dynamical systems," *Computer Methods in Applied Mechanics and Engineering*, vol. 1, no. 1, pp. 1–16, 1972.