International Journal of Non-Linear Mechanics 57 (2013) 146–153

Contents lists available at ScienceDirect

International Journal of Non-Linear Mechanics

journal homepage: www.elsevier.com/locate/nlm

A logarithmic complexity floating frame of reference formulation with interpolating splines for articulated multi-flexible-body dynamics

I.M. Khan [a,*], W. Ahn [b], K.S. Anderson [a], S. De [b]

[a] Computational Dynamics Laboratory, Department of Mechanical Aerospace and Nuclear Engineering, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180, USA

[b] The Center for Modeling, Simulation, and Imaging in Medicine, Department of Mechanical Aerospace and Nuclear Engineering, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180, USA

ARTICLE INFO

Article history:
Received 27 November 2012
Received in revised form
28 June 2013
Accepted 21 July 2013
Available online 27 July 2013

Keywords:
Multi-flexible-body systems
Logarithmic complexity
Divide-and-conquer algorithm
Interpolating splines

ABSTRACT

An interpolating spline-based approach is presented for modeling multi-flexible-body systems in the divide-and-conquer (DCA) scheme. This algorithm uses the floating frame of reference formulation and piecewise spline functions to construct and solve the non-linear equations of motion of the multi-flexible-body system undergoing large rotations and translations. The new approach is compared with the flexible DCA (FDCA) that uses the assumed modes method [1]. The FDCA, in many cases, must resort to sub-structuring to accurately model the deformation of the system. We demonstrate, through numerical examples, that the interpolating spline-based approach is comparable in accuracy and superior in efficiency to the FDCA. The present approach is appropriate for modeling flexible mechanisms with thin 1D bodies undergoing large rotations and translations, including those with irregular shapes. As such, the present approach extends the current capability of the DCA to model deformable systems. The algorithm retains the theoretical logarithmic complexity inherent in the DCA when implemented in parallel.

1. Introduction

Modeling flexibility in the dynamic simulation of multibody systems often becomes unavoidable when its effects are significant, e.g., systems including articulated elastic objects such as bars, beams, shafts, and belts in application areas such as robotics and micro-electromechanical systems (MEMS). The goal of these simulations is to capture the essential dynamics of the system arising from the large overall motion, often regarded as a rigid body motion associated with the kinematic joints between the flexible bodies, and that from the elastic deformation of the individual components of the system. In such simulations, one often encounters situations where computational efficiency as well as model accuracy is important and a balance between the two is required. In many cases, it may be useful to employ multibody methods for computational cost savings, kinematic exactness, and ease in handling large rotations and translations. Using finite element based models may significantly increase computational cost.

A number of approaches have been developed for simulating flexible bodies that also incorporate the effects of gross rigid body motion. Some of the widely adopted approaches include the floating frame of reference formulation (FFR) [2,3], absolute nodal coordinate formulation (for beam and plate type elements) [4,5], and other finite element (FE) based techniques [6–9]. For a brief review and an extensive literature survey on different computational strategies in flexible multibody systems, the reader is referred to [10]. Regardless of the method, computational cost increases with increasing complexity of the system. Therefore, the development of computationally efficient methods which are also accurate has always been an important topic of research in multibody dynamics. In this paper, we present a new multibody method that incorporate interpolating spline in a divide-and-conquer framework. The present algorithm provides an efficient approach for modeling dynamic systems employing articulated rigid and flexible bodies undergoing large rotations and translations.

The divide-and-conquer algorithm (DCA) was introduced by Roy Featherstone [11] as a massively parallel, truly time optimal multibody dynamics algorithm. The DCA is applicable to general multibody systems and achieves logarithmic complexity $O(\log(N_b))$, when implemented on $O(N_b)$ processors [12]. Thus, the DCA is a good candidate for situations where computational efficiency and cost are important. Several variants of the DCA were

* Corresponding author. Tel.: +1 917 445 8333.
E-mail addresses: imad_mahfooz@hotmail.com, khani2@rpi.edu (I.M. Khan), ahnw@rpi.edu (W. Ahn), anderk5@rpi.edu (K.S. Anderson), des@rpi.edu (S. De).
URL: http://www.scer.rpi.edu/cemsim/ (S. De).

0020-7462/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ijnonlinmec.2013.07.002

developed for multibody systems with closed loops [12,13], flexible bodies [1], control problems [14,15], and discontinuous changes arising during simulation [16]. In this paper, we present a DCA based efficient algorithm that utilizes the interpolating spline functions for modeling multi-flexible-body systems. A comparison between the spline-based approach (SDCA) and the assumed modes method scheme is presented. The assumed mode DCA or flexible DCA (FDCA [1]) is appropriate when dealing with bodies with regular geometry such as straight and uniform beams whose mode shapes have been thoroughly investigated in the literature [17]. However, for irregularly shaped beams, the FDCA must resort to sub-structuring in order to adequately capture the deformation of the system [18]. The SDCA approach presented in this paper overcomes this limitation.

Various types of approximating and interpolating spline functions have found applications in a number of areas, ranging from computer graphics to statistics and mechanics. For example, B-splines have been extensively studied in finite element based methods [19,20]. This is due to the computational advantages associated with B-splines including ease of implementation and smoothness. In multibody dynamics, the use of interpolating cubic splines has been studied in [21,22], while others have hinted towards their potential applications in modeling multi-flexible body systems [23,24]. However, a detailed analysis of interpolating spline-based algorithms is lacking, including a study of their performance compared to traditional modeling approaches. Spline functions are appropriate in multibody dynamics because of the relatively small number of nodes in such simulations, as well as their smooth, continuous and interpolating nature at these nodes. The new spline-based algorithm developed here is restricted to modeling multibody systems comprising thin beam like bodies such as ropes, tubes, beams, and polymer chains. Furthermore, the present algorithm can be used in conjunction with other DCA-based algorithms to model systems with different types of rigid and flexible bodies. In this work, we have provided examples corresponding to SDCA for planar mechanisms and 1D bodies. However, the technique may be extended to problems in higher spatial dimensions. The results obtained with the SDCA are compared with the method of superposed assumed modes in FDCA and the numerical complexity of the method is studied. It is demonstrated that the DCA based on interpolating splines provides an alternate and computationally fast method for modeling articulated flexible bodies.

In Section 2, a brief overview of the basic DCA and interpolating splines is presented. The derivation of DCA based spline method is presented in Section 3. Computational complexity of FDCA and SDCA are compared in Section 3. Finally, numerical examples and discussions are presented in Section 4.

2. Theoretical background

In this section, we present a brief overview of the basic DCA and interpolating splines. The computational complexity of the FDCA and SDCA is also presented in this section.

2.1. Basic divide and conquer algorithm

Detailed derivation and analysis of the performance of the DCA can be found in [11,12]. Here, we present the DCA in its basic form. Consider two representative bodies, k and k+1, connected with each other by a joint J^k. Let the points where each generic body interacts with other bodies and the environment, be termed as 'handles'. As an example, consider Fig. 1(a), where H_1^k and H_2^k are the two handles on body k. Similarly, for body k+1, the points H_1^{k+1} and H_2^{k+1} define the position of its handles. For convenience, these handles may correspond to the locations of the joints in a body, for example, the joint J^k can serve as the location for the outward and inward handles for body k and body $k+1$, respectively. The bodies and the joints along with the constraint forces acting on the handles are shown in Fig. 1(a).

There are two main processes in the DCA, the hierarchic assembly and the hierarchic disassembly. In the pre-assembly steps, the equations of motion for each body are formed at its handles. As such, for body k the two-handle equations of motion can be written as

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

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

In the above two equations A_1^k and A_2^k are the 6×1 spatial accelerations of body k at handles H_1^k and H_2^k, respectively. The terms ζ_{ij}^k (i=1,2 and j=1,2,) represent the inverse inertia terms associated with the two handles, whereas ζ_{i3}^k (i=1,2) terms contain all the state dependent accelerations as well as the effects of externally applied loads [12]. The two-handle equations for body k+1 can be given by

$$A_1^{k+1} = \zeta_{11}^{k+1} F_{1c}^{k+1} + \zeta_{12}^{k+1} F_{2c}^{k+1} + \zeta_{13}^{k+1}, \tag{3}$$

$$A_2^{k+1} = \zeta_{21}^{k+1} F_{1c}^{k+1} + \zeta_{22}^{k+1} F_{2c}^{k+1} + \zeta_{23}^{k+1}. \tag{4}$$

The goal of the assembly process is to combine the equations for the successive bodies to form the equations of the resulting assemblies. In case of body k and body k+1, the resulting assembly is illustrated in Fig. 1(b), and the two-handle equations of motion for the resulting assembly

$$A_1^k = \zeta_{11}^{k:k+1} F_{1c}^k + \zeta_{12}^{k:k+1} F_{2c}^{k+1} + \zeta_{13}^{k:k+1}, \tag{5}$$

$$A_2^{k+1} = \zeta_{21}^{k:k+1} F_{1c}^k + \zeta_{22}^{k:k+1} F_{2c}^{k+1} + \zeta_{23}^{k:k+1}. \tag{6}$$

Eqs. (5) and (6) provide the spatial accelerations of the outward handles of the sub-assembly k:k+1. The terms $\zeta_{ij}^{k:k+1}$ (i=1,2 and j=1,2,3) have the same meaning as before, except they now represent the sub-assembly formed by bodies k and $k+1$. Note that the two-handle equations of the sub-assembly are in the same form as the equations of the constituent bodies. This process can be repeated in a hierarchic manner for all successive bodies in the multibody tree. This assembly process starts at the individual body or leaf nodes. The two-handle equations for the pairs of adjacent bodies are combined together to form the equations for the resulting sub-assemblies. This process continues in a hierarchic fashion until the process reaches the root or the primary system node. At this point, the assembly process stops and the two-handle equations of motion for the entire system are obtained. The hierarchic disassembly begins at the primary system node, where by using the boundary conditions, the equations of motion for the last assembly are solved. Using this information, the disassembly process solves the equations of the constituent sub-assemblies. This process continues until the process reaches the individual body nodes. At the end of the disassembly process all unknowns (e.g., spatial constraint forces, modal generalized accelerations, spatial constraint impulses, spatial accelerations, jumps in the spatial velocities) for the bodies at the individual sub-domain level of the binary tree are known. The assembly and disassembly processes are illustrated in Fig. 2.

2.2. Spline interpolation

Spline functions are smooth piecewise interpolating curves that have applications in disciplines including computer graphics, numerical methods and mechanics. Various types of splines, their