

## A Unifying Framework for Rigid Multibody Dynamics and Serial and Parallel Computational Issues

Amir Fijany and Abhinandan Jain

Jet Propulsion Laboratory/California Institute of Technology

### Abstract

In this paper we present a unifying framework for various formulations of the dynamics of open-chain rigid multibody systems and assess their suitability for serial and parallel processing. The framework is based on the derivation of intrinsic, i.e., coordinate-free, equations of the algorithms which provides a suitable abstraction and permits a distinction to be made between the computational redundancy in the intrinsic and extrinsic equations. A set of spatial notation is used which allows the derivation of the various algorithms in a common setting and thus clarifies the relationships among them. The three classes of algorithms viz.,  $O(n)$ ,  $O(n^2)$  and  $O(n^3)$  or the solution of the dynamics problem are investigated. We begin with the derivation of  $O(n^3)$  algorithms based on the explicit computation of the mass matrix and it provides insight into the underlying basis of the  $O(n)$  algorithms. From a computational perspective, the optimal choice of a coordinate frame for the projection of the intrinsic equations is discussed and the serial computational complexity of the different algorithms is evaluated. The three classes of algorithms are also analyzed for suitability for parallel processing. It is shown that the problem belongs to the class of  $NC$  and the time and processor bounds are of  $O(\log_2^2(n))$  and  $O(n^4)$ , respectively. However, the algorithm that achieves the above bounds is not stable. We show that the fastest stable parallel algorithm achieves a computational complexity of  $O(n)$  with  $O(n^4)$ , respectively. However, the algorithm that achieves the above bounds is not stable. We show that the fastest stable parallel algorithm achieves a computational complexity of  $O(n)$  with  $O(n^2)$  processors, and results from the parallelization of the  $O(n^3)$  serial algorithm.