International Symposium on Coupled Methods in Numerical Dynamics
Split, Croatia, September 16–19, 2009

Symbolic Modeling and Analysis of Elastic Multibody Systems

Thomas Kurz and Peter Eberhard

University of Stuttgart, Institute of Engineering and Computational Mechanics,
Pfaffenwaldring 9, 70569 Stuttgart, Germany
[kurz,eberhard]@itm.uni-stuttgart.de

Abstract. The algorithms to set up the equations of motion symbolically for both rigid and elastic multibody systems are presented. In the described program Neweul-M^2 which is using the power of Matlab and Maple, the modeling approach with commands and a graphical user interface are discussed as well as an overview of possibilities for system analysis, control design and optimization is given. A double pendulum is modeled both with rigid and elastic bodies to explain the program features.

Key words: *Symbolic equations of motion, Elastic multibody systems, Research software, Matlab*

1. Introduction

The generation of equations of motion for large multibody systems is a non-trivial task requiring numerous steps during the evaluation of the fundamental relations. Multibody system formalisms are founded on Lagrange's equations of the first or second kind, or the Newton-Euler equations and D'Alembert's or Jourdain's principle, respectively, see e.g. Kane and Levinson [3], Lu [8] and Schiehlen [10]. Regarding the computational procedure, numerical and symbolical formalisms are distinguished. A numerical formalism provides the numbers in the equations of motion required for each time step of the simulation program. In contrary, symbolical formalisms generate the equations of motion only once with a computer code as it would be done with paper and pencil. The advantage is that different values of the system parameters can be inserted in the symbolical equations of motion, but also the structure of the equations can be further utilized. Symbolical formalisms are especially helpful for real-time simulations, optimizations and control design, where the created equations can be used in a flexible way by other programs, too.