Computational methods and implementation for flexible multibody systems

Johannes Gerstmayr[1]

[1] Department of Mechatronics
University of Innsbruck
Technikerstr. 13, 6020 Innsbruck, Austria
johannes.gerstmayr@uibk.ac.at

EXTENDED ABSTRACT

1 Introduction

Engineering problems, and especially problems in multibody system (MBS) dynamics are becoming more complex regarding nonlinearity and dimensionality. A researcher can thus either focus on a small part of the complex problem, use a powerful commercial software, or build the holistic solution upon open source software(s) [7, 4].

This contribution highlights research results that are intended to serve as building blocks to the solution of complex problems, all of these blocks being available as open source software [2, 1]. While a single line of command was sufficient for the computation of a sinusoidal function in the 1960s and for the solution of an ordinary differential equation in the 1990s, it is possible nowadays to create a flexible body or mechanism using a few lines of code.

In many engineering problems, extensibility, compatibility, and merging system components are more important than the ultimate performance of a single simulator. Therefore, the present approach focuses on redundant multibody formalisms which allow almost arbitrary connection of system components. The description of the system is built on coordinates (unknowns), which may be related to differential or algebraic equations. While the coordinates are represented by nodes, equations are defined by computational objects. This approach perfectly fits into the finite element description, but also can represent rigid bodies, kinematic trees as well as abstract mathematical relations. In order to avoid tremendous implementation efforts for interfaces between joints, rigid as well as flexible bodies, the concept of markers is used as interfaces between the single components. To make the framework accessible to researchers, computational tools require an accessible software interface. Such interfaces ideally dock to larger software ecosystems, such as Python and Anaconda, including artificial intelligence (AI).

2 Flexible bodies

Methods for flexible bodies in multibody systems have been developed in order to be able to achieve highly accurate simulation of complex-shaped bodies undergoing small deformations, and for highly slender structures which can undergo large deformations as well. Due to the specific formulations and reduced order modeling, the computational methods are superior to conventional finite element methods when it comes to dynamic simulation. The multibody approach can also be applied to large scale compliant structures within programmable matter [6].

For highly slender structures, geometrically exact beams and shells have been developed since the 1980s. Due to the intrinsic numerical problems of large rotations when solved by conventional time integration methods, the so-called absolute nodal coordinate formulation (ANCF) has been developed since 25 years [5]. While original elements showed severe Poisson and shear locking effects, improvements have been proposed in order to obtain formulations with high performance and accuracy. The developed methods are superior to rotation-parameter-based methods in case of higher demands on continuity in nodes and in case of special material laws, which cannot be incorporated into geometrically exact beams straight away. It could also be shown that ANCF elements exactly converge to geometrically exact beams if according elastic force laws are employed. The main achievements on ANCF will be shortly highlighted during the talk.

For solid bodies, the floating frame of reference formulation (FFRF) has been developed since the 1980s. While the equations of motion are a standard in textbooks on flexible MBS, the computer implementation has been tedious for general spatial solid finite elements. Just recently, a highly simplified approach could be shown, which allows a straight-forward computation of the FFRF equations of motion, just using mass and stiffness matrices, as well as nodal reference coordinates. The resulting highly nonlinear equations of motion just require some dozens of lines of code with the great advantage to use them in any framework or to extend them to more complex cases. A complete framework based on this formulation, including a novel formulation of interface equations will be shown together with numerical examples.

3 Computational tools

For reliable and efficient time integration of multibody systems – the classical forward simulation, there exist established solution methods. In cases where the multibody system is presented by a set of non-stiff ordinary differential equations, explicit solvers are optimal as they deliver high accuracy at low computational costs and few lines of code for implementation. In the case of stiff or differential-algebraic equations, implicit solvers have been developed, many of them based on the trapezoidal rule, such as the index 3 generalized-α solver. However, in case that no numerical damping shall be applied, index 2 integrators have been proven to be useful. While Euler parameters have become a standard in multibody system dynamics, Lie group methods have been developed in order to avoid redundant rotation parameters and the related overhead, which will be compared to each other.

The field of engineering science is driven by the availability of easy accessible but still powerful computational tools. While FORTRAN may be even seen as a first tool, software packages such as MATLAB have enabled researchers to efficiently develop new multibody methods in short time, however, at a proprietary level. Due to the openness, Python and similar languages have been driven by the open source community and gained superior features in fields, mentioning exemplary numerical computation (SciPy), computer vision (OpenCV), robotics (ROS) as well as machine learning (PyTorch). In the same manner, multibody dynamics tools can be embedded in such environments to be able to solve much larger problems than just forward simulation. In order to achieve efficiency up to a multithreaded implementation and for processing of large data sets, C++ is utilized within core functions. However, for easy interfacing and catching of user errors, the Python language is used. Common Python packages allow to run the resulting package on single-board computers (Raspberry Pi) as well as on MPI-parallelized supercomputers.

Often, the pure forward simulation is not the primary focus of engineers. They are rather interested in sensitivities of parameters to oscillations or stresses, or on extremal cases, e.g., maximum speed, deformation or drive torque in a mechanism. Therefore, a simple way to provide higher functionality upon classical static and dynamic computations, appropriate interfaces to parallelized parameter variation, optimization or sensitivity analysis is presented with test cases.

Finally, having computational tools in Python at hand allows to close the gap between AI and MBS methods. It will be shown how multibody system methods can be integrated with established AI methods, such as reinforcement learning (RL). The direct integration of parameterized multibody dynamic models into Python allows to close the loop with RL algorithms. It will be shown how common parallelization techniques, such as multithreading and MPI can be used on such platforms, in order to perform not only one training and evaluation, but to investigate randomized training runs, showing the reliability and stability of the algorithms in engineering problems. Furthermore, the trustworthiness of trained models is evaluated systematically.

While open source allows users to easily install and use existing tools, the underlying functionality and the problem setup may be completely viewed, which is needed for real open access publications. Furthermore, open source tools may be extended by expert users. However, an interesting and promising feature of open source codes comes with Generative Pretrained Transformers, such as GPT-4 [3], which allows to assist users of open source software in answering questions or even providing code (fragments) with astonishing level of completeness.

4 Conclusions and outlook

This presentation highlights achievements in computational methods for rigid and flexible multibody systems during the past 25 years. Therein a main focus has been the accessibility and the easiness of reliable and powerful methods. Advances are highlighted in the presentation and future directions will be proposed.

Acknowledgments

In order to achieve the presented results, the collaboration with PhD students and research colleagues during the past 25 years is gratefully acknowledged.

References

[1] Gerstmayr, J.: Exudyn GitHub repository. University of Innsbruck, Department of Mechatronics. https://github.com/jgerstmayr/EXUDYN (accessed on March 8, 2023) (2023)

[2] Gerstmayr, J.: HOTINT – A C++ Environment for the simulation of multibody dynamics systems and finite elements. In: Arczewski, K., Fraczek, J., Wojtyra, M. (eds.) Proceedings of the Multibody Dynamics 2009 Eccomas Thematic Conference (2009)

[3] OpenAI: ChatGPT, OpenAI Inc. / OpenAI LP. https://chat.openai.com (accessed on March 8, 2023) (2023)

[4] Masarati, P., Morandini, M., Mantegazza, P.: An Efficient Formulation for General-Purpose Multibody/Multiphysics Analysis. Journal of Computational and Nonlinear Dynamics **9**(4) (2014). 10.1115/1.4025628. 041001

[5] K. Otsuka, K. Makihara, H. Sugiyama, Recent Advances in the Absolute Nodal Coordinate Formulation: Literature Review From 2012 to 2020., ASME. J. Comput. Nonlinear Dynam., Vol. 17(8): 080803, 2022.

[6] Pieber, M., Gerstmayr, J.: Six-Bar Linkages With Compliant Mechanisms for an Adaptive Robot. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. 10: 44th Mechanisms and Robotics Conference (MR) (2020). 10.1115/DETC2020-22546. DETC2020-22546

[7] Mazhar, H., Heyn, T., Pazouki, A., Melanz, D., Seidl, A., Bartholomew, A., Tasora, A., Negrut, D.: CHRONO: A parallel multi-physics library for rigid-body, flexible-body, and fluid dynamics. Mechanical Sciences 4, 49–64 (2013). 10.5194/ms-4-49-2013