PAMM · Proc. Appl. Math. Mech. **2**, 130–131 (2003) / **DOI** 10.1002/pamm.200310050

HELM, ALEXANDER; HARDT, MICHAEL; VON STRYK, OSKAR; HÖPLER, ROBERT

Development of a Toolbox for Model-Based Real-Time Simulation and Analysis of Legged Robots

Multibody systems such as legged robots require sophisticated and efficient methods for their modeling, control and simulation. This paper discusses the development of a software library based on modern tools such as C++, OpenGL and XML for highly efficient dynamics modeling. A primary focus lies on modularity permitting its easy extensibility in connection with different actuation and contact models, optimization algorithms, localized and centralized on-line control schemes as well as animation and simulation environments.

1. Modeling of Legged Systems

The toolbox is based on concepts presented in [1] for object-oriented robot modeling in a real-time environment with its extension to a general multibody modeling scheme. This new toolbox is intended to form the core for multibody systems design, analysis, and control such as in the construction of a legged robot, the performance analysis of triathletes, or the running control of a soccer-playing four-legged robot which all require sophisticated simulation and optimization tools [5]. Though different biomechanical applications may rely on widely different dynamic models, the underlying recursive structure remains the same. One of the long-term goals is to enable the user to easily interchange models of muscles and motors, friction and ground terrain models, or ambient fluid dynamics to perform simulations as well as kinematic and dynamic optimization. To achieve these goals, modularity and efficiency have to be considered.

Depending upon the application, a high degree of modeling modularity is invaluable when dealing with complex models such as walking systems with a changing number of mechanical degrees of freedom, depending upon contact conditions. Advanced walking systems require the implementation of various actuation models like motors, gears or even (artificial) muscles. Operation in an dynamic environment also means there is the need for modeling various exterior forces, e. g. payloads, ambient wind or water and multiple contact forces.

Efficiency is of paramount importance for the above applications. Time efficiency is crucial for real-time simulation and on-line model-based control of walking systems. Memory efficiency must also be considered, in particular in possible application areas such as embedded systems which usually have very limited resources (including memory and computing power). There is virtually no upper limit to the desired level of modeling complexity for these systems. By achieving greater efficiency, the number of systems that may be considered can be substantially increased. At the same time a flexible and modular structure is desired where, for example, parameters may be changed on-line without restart and recompilation. The chosen recursive algorithmic structure satisfies these characteristics.

2. Multibody Dynamic Algorithms

The equations of motion of a legged system modeled as a rigid multi-body system (MBS) can be written in the well-known form

$$\mathcal{M}(q)\ddot{q} + \mathcal{C}(q,\dot{q}) + \mathcal{G}(q) = u + J_{\mathcal{C}}(q)^T f_{\mathcal{C}}$$

where $\mathcal{M}$ is the mass-inertia matrix, q is the generalized position states of the walking system, $\mathcal{C}$ denotes the Coriolis forces and $\mathcal{G}$ gravitational forces. The vector u represents the applied forces, ${J_{\mathcal{C}}}^T$ is the transpose of the constraint Jacobian, and $f_{\mathcal{C}}$ contains the contact forces.

Several approaches exist in literature for the solution of the forward dynamics problem. Symbolic methods [6] can construct efficient, closed form dynamics with $\mathcal{O}(N)$ complexity for legged systems through symbolic manipulation of the above equation. In stages of system design and control, often frequent changes in the models of the robot and its environment are necessary which makes these approaches impractical. The Composite Rigid Body Algorithm (CRBA) [7] and the Articulated Body Algorithm (ABA) [4] both satisfy the need for modularity, while the former is more efficient for small MBS (systems with 9 degrees of freedom or less) though of $\mathcal{O}(N^3)$ complexity, while the latter is $\mathcal{O}(N)$, more efficient for complex systems, and well suited for larger robotic systems.

ABA is a recursive algorithm which exploits the numerical advantage of formulating the rigid body dynamics in an alternative manner based on intermediate mass-inertia descriptions of parts of the tree-structured system. Instead of expressing the spatial forces acting at each link as the product of link mass and link acceleration plus bias forces, masses and accelerations of ‘articulated’ chains are used. The standard definition of forces from the Newton-Euler description is $f = M\dot{V} + \beta$ where $f = \text{vec}\{f_i\} \in \mathbb{R}^{6N\times 1}$ contains the 6-dimensional spatial (rotational + linear) forces of each link i, $M = \text{diag}\{M_i\} \in \mathbb{R}^{6N\times 6N}$ contain the system’s spatial inertias, $\dot{V} = \text{vec}\{\dot{V}_i\} \in \mathbb{R}^{6N\times 1}$ are the spatial accelerations, while β contains bias forces. The forces may be alternatively described using the articulated body spatial inertia P, $f = P\dot{V} + z$. The articulated body may be considered as the ‘floppy chain’ of outboard links in the tree-structured multi-body system. Featherstone made this discovery and also introduced the 6-dimensional spatial notation [3]. Rodriguez and Jain made the association to Kalman filtering which permitted them to construct an $\mathcal{O}(N)$ Cholesky type factorization (LDL^T) of the mass-inertia matrix. Their ‘innovations factorization’ of the mass-inertia matrix $\mathcal{M}$ written as $\mathcal{M} = [I + K\phi H]^T D[I + K\phi H]$ in terms of spatial operators has as closed form inverse

$$\mathcal{M}^{-1} = [I - K\psi H]D^{-1}[I - K\psi H]^T$$

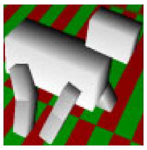
which may be derived from algebraic identities [4]. The entire forward dynamics algorithm in the unconstrained setting may be conducted in three sweeps of the tree-structured multibody chain (base → tip → base → tip).

The modeling paradigm for the investigated legged systems is based on that presented in [1]. It is well-suited for an efficient implementation of recursive dynamics algorithms such as the ABA. It relies on a port-based formulation for the definition of a rigid MBS model which may consist of components like bodies, various joints, actuators, contacts and other constraints. Within the modeling paradigm, interconnections between MBS entities occurring through certain ports called *connectors* define the model topology. These serve as a basis for the implementation of efficient data transfer between the software components performing the dynamics computations, as have been similarly implemented in a C++ recursive inverse dynamics algorithm for on-line control purposes [1].

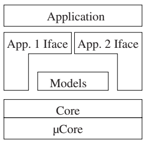
3. *RoDynA*: *Ro*bot *Dyn*amics *A*nalysis

RoDynA combines concepts from [1] and [2] into a new toolbox for robot dynamics and analysis. Its capabilities extend to the ability to model free-floating robots and legged systems. A primary target of *RoDynA* is to provide easy integration of new algorithms, e. g. dynamic contact models or exact sensitivities of the dynamics. On a higher level new interfaces especially for numerical integrators, optimizers, simulators, graphical user interfaces and controllers shall guarantee its extensibility and ease of use. In addition, *RoDynA* is designed for real-time simulation and on-line control.

With respect to our research, *RoDynA* will have a suitable interface for many applications. The interface figure gives some idea of the interface. The *μCore* is the part that will not be changed. The *Core* provides basic functionality common to almost every application and model. *Models* are new entities like muscles. *App. n Iface* represents interfaces between certain models and applications. It must be assured that it is easy to add additional functionality even on the lowest level. On the other hand, higher level functions should not be affected by these changes as long as the extensions are not in use. For debugging and analysis there will be some manner of logging and validation to retain data integrity. This concept is similar to the transaction concept of databases. The interface is implemented in pure C++ thus enabling fast communication and low memory consumption. C++ is more advantageous than C because it provides more modularity through its object-orientation. The price to pay for this are platform and compiler dependencies among others. Additionally, *RoDynA* will have an object naming and browsing facility.



Simulation of a Sony Aibo



Interface Design

4. References

1 Robert Höpler and Pieter J. Mosterman: “Model Integrated Computing in Robot Control to Synthesize Real-time Embedded Code,” Proceedings of the IEEE Conference on Control Applications, Mexico City, pp. 767–772, 2001.

2 Michael Hardt: “Multibody Dynamical Algorithms, Numerical Optimal Control, with Detailed Studies in the Control of Jet Engine Compressors and Biped Walking,” Electrical and Computer Engineering, UC San Diego, June 1999.

3 R. Featherstone: *Robot Dynamics Algorithms*, Kluwer Academic Publishers, 1987.

4 G. Rodriguez, K. Kreutz-Delgado, and A. Jain “A Spatial Operator Algebra for Manipulator Modeling and Control,” The International Journal of Robotics Research, Vol. 10, No. 4, pp. 371–381, Aug. 1991.

5 M. Hardt, O. von Stryk: “Towards optimal hybrid control solutions for gait patterns of a quadruped,”CLAWAR: Int’l Conf. on Climbing and Walking Robots , Madrid, pp. 385-392, October 2000.

6 H. G. Kwatny and G. L. Blankenship: “Symbolic Construction of Models for Multibody Dynamics,” IEEE Transactions on Robotics & Automation, Vol. 11, No. 2, pp. 271-281, April 1995.

7 M. Walker and D. Orin: “Efficient Dynamic Computer Simulation of Robotic Mechanisms,” ASME Journal of Dynamic Systems, Measurement and Control, Vol. 104, No. 3, pp. 205-211, 1982.

Alexander Helm, Dr. Michael Hardt, Prof. Dr. Oskar von Stryk, TU Darmstadt,
(helm|hardt|stryk)@sim.tu-darmstadt.de
Robert Höpler, DLR Oberpfaffenhofen, Robert.Hoepler@dlr.de