Spatial Operator Algebra Perspective for Computational Multibody Dynamics

Abhinandan Jain (Abhi.Jainjpl.nasa.gov)
Jet Propulsion Laboratory/California Institute of Technology

This paper will provide an overview of the Spatial Operator Algebra (SOA) framework for multibody dynamics. Topics covered will range from its theoretical underpinnings, to its use for computational dynamics and implementations for modeling and simulation in a variety of domains.

It is a well known fact that the complexity of the dynamical equations of motion of multibody systems grows rapidly with system size. Spatial operators provide mathematical constructs for concisely representing and manipulating these dynamic expressions and for managing their complexity. One of its key innovations is that it eliminates mathematical complexity. It reduces the number of symbols that a human analyst must see and understand by several orders of magnitude. While complexity is eliminated, no information about the system is lost in the Spatial Operator approach, as the spatial operators still contain precise information about the system. However, this information is organized and presented to the analyst using much fewer symbols. Spatial operators are based on certain mathematical parallels between the domains of multibody dynamics and optimal estimation theory. That the Kalman filter embedded in the Spatial Operator Algebra solves computational bottlenecks in classical mechanics is a very surprising scientific discovery, requiring a deep understanding of both classical mechanics and filtering methods. For instance, the solution to the equations of motion at a time instant is mathematically equivalent to solving an optimal smoothing problem over a finite interval.

The SOA have been systematically using the Spatial Operator Algebra to solve a number of important research problems in robotics research: 1) analytical inversion of the manipulator mass matrix required to conduct robot dynamics simulation; 2) non-interacting manipulator control, in which the control system at each of the joints is independent of all of the other joints; 3) control of underactuated manipulators, in which not all of the degrees of freedom are actively controlled; and 4) dual arm manipulation in which two or more arm-like systems work together to move a commonly held object; 5) operational space dynamics and control where the robot control problem is posed in terms of the end-effector task space; 6) explicit linearization and sensitivity models for the dynamics equations of motion; 7) diagonalizing quasi-velocity coordinate representations of the system dynamics etc. Many of the spatial operators, those requiring inversion of the system mass matrix for example, are mechanized by spatially recursive Kalman filtering algorithms, quite popular in digital filtering and signal processing.

A relatively less understood aspect of spatial operators is that mathematical description and analysis applies to not just rigid multibody systems but to ones with body and hinge flexibility. While the details of the component operators depend on the specific properties of the mechanical system, the spatial operator mathematics remains unchanged. The spatial operators thus serve as unifying mathematics and analysis constructs for the full domain of multibody systems.

The spatial operator approach and computational algorithms have found application in a number of key areas. They have formed the basis for the DARTS/Dshell family of high-fidelity modeling and simulation software and tools for a number of NASA's space missions.

The DARTS flexible multibody dynamics software forms the basis of the Dshell simulation toolkit which serves as the foundation of a the simulators for a number of domains. Such simulators include the ROAMS simulator for rover vehicles for planetary surface exploration and the DSENDS simulator for entry, descent and landing on planetary surfaces.

The Spatial Operator Algebra is being used to study the dynamics of large molecular structures. Areas of investigation include the structural and functional relationships of proteins and enzymes; protein folding mechanisms and pathways; new drug design; and design and study of catalysts and polymers. The fundamental technical problem being addressed is the global and local dynamical behavior a complex collection of many atoms joined together by interactive forces. The efficiency of the dynamics algorithms in the Spatial Operator Algebra enable the accurately detailed study of much larger systems than could be studied with previous methods.

The Spatial Operator Algebra is easily implemented using modern software development methods. At any given level in the hierarchy, the computer automatically decomposes each spatial operator into a set of a few more detailed operations at the next lower level in the hierarchy. There is a corresponding decomposition in the software architecture. This process of decomposition can be viewed as that of "smart" compilation, in the sense that the compiler is made smart by built-in knowledge about the system dynamical model embedded in the spatial operators. The software modules which constitute the architecture are simple, standardized, and easily debugged.

This programming approach achieves a very high level of abstraction. The number of symbols visible to the analyst at any level in the hierarchy is very small. This means that the corresponding computer programs are also very simple. The programs are modular and map to a modular software architecture. The programs are also to a large extent system-independent, in the sense that going from one system to another is easy to do. Although the programs are simple, computational efficiency is not lost. Embedded in the programs are very efficient computational algorithms. We are developing the PyCraft computational workbench to solve this problem. PyCraft allows the easy implementation and evaluation of sophisticated algorithms developed using SOA. PyCraft provides an interactive environment where users can directly utiles high-level SOA operator expression to describe and execute various dynamics quantities and computations interactively. While PyCraft continues to evolve, it already supports a large number of dynamics algorithms for computational experiments.