Simulation Modelling Practice and Theory 18 (2010) 63–86

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

Object-oriented modelling for spacecraft dynamics: Tools and applications

T. Pulecchi, F. Casella *, M. Lovera

Dipartimento di Elettronica e Informazione, Politecnico di Milano, 20133 Milano, Italy

ARTICLE INFO

Article history:
Received 22 December 2008
Accepted 24 September 2009
Available online 2 October 2009

Keywords:
Object-oriented modelling
Spacecraft dynamics
Orbital control
Attitude control

ABSTRACT

The development process for spacecraft control systems relies heavily on modelling and simulation tools for spacecraft dynamics. For this reason, there is an increasing need for adequate design tools in order to cope efficiently with tightening budgets for space missions. The paper discusses the main issues related to the modelling and simulation of satellite dynamics for control purposes, and then presents an object-oriented modelling framework, implemented as a Modelica library. The proposed approach allows a unified approach to a range of problems spanning from initial mission design and actuator sizing phases, down to detailed closed-loop simulation of the control system, including realistic models of sensors and actuators. It also promotes the reuse of modelling knowledge among similar missions, thus minimizing the design effort for any new project. The proposed framework and the Modelica library are demonstrated by several illustrative case studies.

1. Introduction

The safe and satisfactory operation of a satellite, in terms of its mission objectives, is strongly related to the performance level of its on-board attitude and orbit control systems, which provide the ability to maintain a desired orientation in space (or, e.g., carry out predefined attitude maneuvers) and track a desired, nominal orbit in spite of the presence of external disturbances. In addition, the recent trend towards missions based on constellations or formations of small satellites has led to the formulation of even more complex control problems, related to the relative motion (both in terms of attitude and position) of multiple spacecraft at the same time. This has resulted in an increasing need for efficient tools in every domain involved in spacecraft design, particularly in the area of control-oriented modelling and simulation. Specific tools are required for the design of both the system architecture and the Attitude and Orbit Control System (AOCS), bearing in mind the principles of reusability, flexibility and modularity.

The main goal of such tools should be to provide the designer with a *unified environment*, satisfying the following requirements:

- to enable the rapid development, test and integration of new space systems in a unified, multi-physics framework;
- to allow the support and management of a variety of spacecraft configurations and space environment descriptions covering all the phases of mission development;
- to permit the reuse and customization of models across various missions;
- to provide reusable models, interconnecting which complex system networks can be obtained;
- to handle multi-spacecraft simulations;
- to ensure visibility into the simulator models and state variables.

* Corresponding author. Tel.: +39 02 2399 3465; fax: +39 02 2399 3412.
E-mail addresses: casella@elet.polimi.it (F. Casella), lovera@elet.polimi.it (M. Lovera).

1569-190X/$ - see front matter
doi:10.1016/j.simpat.2009.09.010

A number of commercial tools are available to support one or more of the above mentioned requirements in the development of AOCS subsystems; however, none of them seems capable to provide the complete coverage of the whole development cycle in a sufficiently flexible way.

The systematic approach to modelling and simulation offered by modern a-causal, equation-based object-oriented languages such as Modelica (see [17,6,5]) allows to develop spacecraft simulation tools, the use of which is made much more efficient by the very nature of the selected modelling approach. Note, in passing, that there is an increasing interest for multi-domain problems in the spacecraft control design community (see, e.g., [29]), an area which would benefit from the availability of simulation tools based on the object-oriented approach.

The development of simulation tools for satellite attitude and orbit dynamics within the object-oriented paradigm has been the subject of previous work (see [35], where an overview of the existing tools for AOCS modelling is presented). The use of Modelica for aerospace applications has recently led to the development of a library for flight dynamics (see [19]), while some preliminary results in the development of a Modelica spacecraft modelling library have been presented in [11]. More recently, the model components presented in the cited reference have been revised in order to take advantage of the Modelica Multibody library (see [21]) which turns out to be extremely suitable to serve as a basis for the development of models for the mechanical parts of spacecraft models. In addition, exploiting Modelica's advanced features (see, e.g., [22]) it has been possible to redesign the library [11] in order to improve the flexibility and the usability of the simulation tools. Furthermore, a recent extension of the Multibody library (see [31]) is proving specially beneficial for the simulation of spacecraft with flexible appendages (see also [30]), while some related results on the development of high accuracy methods for the simulation of orbit dynamics have been presented in [2].

In view of the above discussion, the aim of this paper is to present a unified modelling approach to spacecraft dynamics modelling and simulation for control purposes based on object-oriented modelling concepts, and the Modelica library Space Flight Dynamics that has been developed to implement it. The capabilities of the proposed approach are then demonstrated by presenting a number of case studies corresponding to applied research work developed in the last few years. In particular, one of the considered applications deals with the development of a simulator and the preliminary design of control laws for attitude regulation for the satellites of the SWARM ESA mission (see [7]).

The paper is organised as follows: a short introduction to the Modelica language will be given in Section 2; the proposed approach to spacecraft modelling will be described in Sections 3 and 4, together with the new models and components developed specifically to implement this framework; the results obtained in three case studies will be presented and discussed in Section 5. More precisely, in Section 5.1 the problem of determining disturbance torques using dynamic inversion will be discussed, in Section 5.2 the modelling and simulation of spacecraft with flexible appendages will be addressed and finally in Section 5.3 the development of a control-oriented simulation model for the SWARM ESA mission will be presented.

2. The Modelica object-oriented modelling language

2.1. Overview of Modelica

Modelica [15,34,5] is a language for hierarchical, equation-based, object-oriented modelling of physical systems, developed by the *Modelica Association* [16]. The main features of the language which are relevant within the scope of this work are summarized here.

Modelica is an object-oriented language, supporting encapsulation, composition and inheritance. These features facilitate model development and update. Elementary models of physical elements are defined in a declarative way by their constitutive equations, and their interface with the outer world is described by physical ports (or connectors) without any implied causality, rather than by writing assignments relating inputs to outputs. This makes the description of physical systems much more flexible and natural than it is possible with causal or block-oriented modelling languages, which describe elementary models by means of assignments relating inputs to outputs, or by directly writing simulation code using procedural languages such as C or FORTRAN. Complex models can then be built by connecting elementary models through their ports; since the ports are a-causal, any connection which is physically meaningful is allowed without restrictions. The Modelica language includes graphical annotations, which allow to use graphical user interfaces (such as the one provided by the tool Dymola) to select components from a library, drag them into a diagram, connect them, and set their parameters, thus making the process of model development highly intuitive for end users.

Since models are described in terms of generic differential-algebraic equations, components which define interactions between different physical domains (e.g., electrical and mechanical) can be combined without any restriction, to form multi-physics models.

The definition of suitable interfaces is at the very heart of the development of any Modelica library, and is of great importance in order to facilitate model reusability and standardization. An interface fully defines how a given component can be connected to the outside world, through either physical or signal ports. On that account, once a component interface has been designed, it seems obvious practice to develop each new model for that component by extending its interface. In order to minimize the development time, Modelica goes a step further, allowing the definition and reuse of intermediate