High-Accuracy Orbital Dynamics Simulation through Keplerian and Equinoctial Parameters

Francesco Casella Marco Lovera
Dipartimento di Elettronica e Informazione
Politecnico di Milano
Piazza Leonardo da Vinci 32, 20133 Milano, Italy

Abstract

In the last few years a Modelica library for spacecraft modelling and simulation has been developed, on the basis of the Modelica Multibody Library. The aim of this paper is to demonstrate improvements in terms of simulation accuracy and efficiency which can be obtained by using Keplerian or Equinoctial parameters instead of Cartesian coordinates as state variables in the spacecraft model. The rigid body model of the standard MultiBody library is extended by adding the equations defining a transformation of the body center-of-mass coodinates from Keplerian and Equinoctial parameters to Cartesian coordinates, and by setting the former as preferred states, instead of the latter. The remaining parts of the model, including the model of the gravitational field, are left untouched, thus ensuring maximum re-usability of third-party code. The results shown in the paper demonstrate the superior accuracy and speed of computation in the reference case of a point-mass gravity field.
Keywords: Spacecraft dynamics; Orbit dynamics; Numerical integration; State selection.

1 Introduction

The Modelica Spacecraft Dynamics Library ([6, 7, 10]) is a set of models (based on the already existing and well known Multibody Library, see [9]) which is currently being developed with the aim of providing an advanced modelling and simulation tool capable of supporting control system analysis and design activities for both spacecraft attitude and orbit dynamics. The main motivation for the development of the library is given by the significant benefits that the adoption of a systematic approach to modelling and simulation, based on modern a-causal object-oriented languages such as Modelica, can give to the design process of such advanced control systems.

At the present stage, the library encompasses all the necessary utilities in order to ready a reliable and quick-to-use scenario for a generic space mission, providing a wide choice of most commonly used models for AOCS sensors, actuators and controls. The library's model reusability is such that, as new missions are conceived, the library can be used as a base upon which readily and easily build a simulator. This goal can be achieved simply by interconnecting the standard library objects, possibly with new components purposely designed to cope with specific mission requirements, regardless of space mission scenario in terms of either mission environment (e.g., planet Earth, Mars, solar system), spacecraft configuration or embarked on-board systems (e.g., sensors, actuators, control algorithms).

More precisely, the generic spacecraft simulator consists of an Extended World model and one or more Spacecraft models. The Extended World model is an extension of Modelica.MultiBody.World which provides all the functions needed for a complete representation of the space environment as seen by a spacecraft: gravitational and geomagnetic field models, atmospheric models, solar radiation models. Such an extension to the basic World model as originally provided in the MultiBody library plays a major role in the realistic simulation of the dynamics of a spacecraft as the linear and angular motion of a satellite are significantly influenced by its interaction with the space environment. The Spacecraft model, on the other hand, is a completely reconfigurable spacecraft including components to describe the actual spacecraft dynamics, the attitude/orbit control sensors and actuators and the relevant control laws. In this paper we are specifically concerned with the Spacecraft model; this component has been defined by extending the already available standard model Modelica.Mechanics.MultiBody.Parts.Body. The main modifications reside in the selectable evaluation of the in-

teractions between the spacecraft and the space environment and on the additional initialization option for the simulation via selection of a specific orbit for the spacecraft. The main drawback associated with the adoption of the standard Body model as the core of the Spacecraft model is related to the intrinsic use this component makes of the Cartesian coordinates in the World reference frame for the state variables associated with the motion of the Body's center of mass. Indeed, for spacecraft work it is well known that significant benefits, both in terms of simulation accuracy and computational performance, can be obtained by using different choices of state variables, such as Keplerian and Equinoctial parameters (see, e.g., [11, 8]).
Therefore, the aims of this paper, which extends preliminary results presented in [2] are the following:

- to demonstrate improvements in terms of simulation accuracy and efficiency which can be obtained by using Keplerian and Equinoctial parameters instead of Cartesian coordinates as state variables in the spacecraft model;
- to illustrate how Keplerian and Equinoctial parameters can be included in the existing multibody spacecraft model by exploiting the object-oriented features of the Modelica language and the symbolic manipulation capability of Modelica tools.

The paper is organised as follows: first an overview of the available choices for the state representation of satellite orbits is given in Section 2; subsequently, the use of Keplerian and Equinoctial orbital elements for the simulation of orbit dynamics will be described in Section 3, while the corresponding Modelica implementation will be outlined in Section 4 and the results obtained in the implementation and application of the proposed approach to the simulation of a Low Earth and Geostationary orbits will be presented and discussed in Section 5.

2 Satellite State Representations

The state of the center of mass of a satellite in space needs six quantities to be defined. These quantities may take on many equivalent forms. Whatever the form, we call the collection of these quantities either a state vector (usually associated with position and velocity vectors) or a set of elements called orbital elements (typically used with scalar magnitude and angular representations of the orbit). Either set of quantities is referenced to a particular reference frame and completely specifies the two-body orbit from a complete set of initial conditions for solving an initial value problem class of differential equations.
In the following subsections, we will deal with spacecraft subject only to the gravitational attraction of the Earth considered as a point mass (*unperturbed Keplerian conditions*) and we will refer mainly to the Earth Centered Inertial reference axes (ECI), defined as follows. The origin of these axes is in the Earth's centre. The X-axis is parallel to the line of nodes. The Z-axis is parallel to the Earth's geographic north-south axis and pointing north. The Y-axis completes the right-handed orthogonal triad.

2.1 Position and Velocity Coordinates

In the ECI reference frame, the position and velocity vectors of a spacecraft influenced only by the gravitational attraction of the Earth considered with punctiform mass will be denotated as follows

$$r = \begin{bmatrix} x & y & z \end{bmatrix}^T, \tag{1}$$

$$v = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T = \frac{dr}{dt}. \tag{2}$$

The acceleration of such a spacecraft satisfies the equation of two-body motion

$$\frac{d^2 r}{dt^2} = -GM_{\oplus}\frac{r}{\|r\|^3} \tag{3}$$

where $\mu = GM_{\oplus}$ is the gravitational coefficient of the Earth. A particular solution of this second order vector differential equation is called an orbit that can be elliptic or parabolic or hyperbolic, depending on the initial values of the spacecraft position and velocity vectors $r(t_0)$ and $v(t_0)$. Only circular and elliptic trajectories are considered in this study.
The state representation by position and velocity of a spacecraft in unperturbed Keplerian conditions is

$$x_{ECI} = \begin{bmatrix} r^T & v^T \end{bmatrix}^T \tag{4}$$

at a given time t. Time t is always associated with a state vector and it is often considered as a seventh component. A time used as reference for the state vector or orbital elements is called the epoch.

2.2 Classical Orbital Elements

The most common element set used to describe elliptical orbits (including circular orbits) are the classical orbital elements (COEs), also called the Keplerian parameters, which are described in the sequel of this Section. The COEs are defined as follows:

- a : semi-major axis, [m];
- n : mean motion, [rad/s]
- e : eccentricity, [dimensionless];
- i : inclination, [rad];
- Ω : right ascension of the ascending node, [rad];
- ω : argument of perigee, [rad];
- ν : true anomaly, [rad];
- E : eccentric anomaly, [rad];
- M : mean anomaly, [rad];

(see Figures 1 and 2). The definitions of the COEs are referenced to the ECI frame. The semi-major axis a specifies the size of the orbit. Alternatively, the mean motion

$$n = \sqrt{\frac{GM_{\oplus}}{a^3}} \tag{5}$$

can be used to specify the size.
The eccentricity e specifies the shape of the ellipse. It is the magnitude of the eccentricity vector, which points toward the perigee along the line of apsis.
The inclination i specifies the tilt of the orbit plane. It is defined as the angle between the angular momentum vector $h = r \times v$ and the unit vector Z.
The right ascension of the ascending node Ω is the angle from the positive X axis to the node vector n pointing toward the ascending node, that is the point on the equatorial plane where the orbit crosses from south to north. The argument of perigee ω is measured from the ascending node to the perigee, i.e., to the eccentricity vector e pointing towards the perigee.
The eccentric anomaly E is defined on the auxiliary circle of radius a, that can be drawn around the elliptical orbit, as shown in Figure 2. Finally, the mean anomaly M is defined as $M = n(t - t_p)$, where t_p denotes the time of perigee passage, i.e., the instant at which the eccentric anomaly vanishes. As is apparent from its definition, the mean anomaly for an ideal Keplerian orbit increases uniformly over time. E and M are related by the well known Kepler equation

$$E - e\sin(E) = M. \tag{6}$$

In this work, satellite state representation in terms of classical orbital elements (Keplerian parameters) will be denoted as

$$x_{COE} = \begin{bmatrix} a & e & i & \Omega & \omega & M \end{bmatrix}^T \tag{7}$$

with the implicit choice of adopting M as a parameter to represent the spacecraft anomaly; the advantages and disadvantages of this choice will be discussed in the following.

2.3 Equinoctial Orbital Elements

COEs suffer from two main singularities. The first is when the orbit is circular, i.e., when the eccentricity is zero ($e = 0$). In this case the line of apsis is undefined and also the argument of perigee ω. The second occurs when the orbit is equatorial, i.e., when the inclination is zero ($i = 0$). In this case the ascending node is undefined and also the right ascension of the ascending node Ω. See Figure 1.

It is nevertheless possible to define the true, eccentric and mean longitude (L, K and l, respectively) as

$$L = \omega + \Omega + \nu, \tag{8}$$
$$K = \omega + \Omega + E, \tag{9}$$
$$l = \omega + \Omega + M; \tag{10}$$

these quantities remain well-defined also in the singular cases of circular and/or equatorial orbits.

The equinoctial orbital elements (EOEs) avoid the singularities encountered when using the classical orbital elements. EOEs were originally developed by Lagrange in 1774. Their definitions in terms of Keplerian elements are given by the following equations

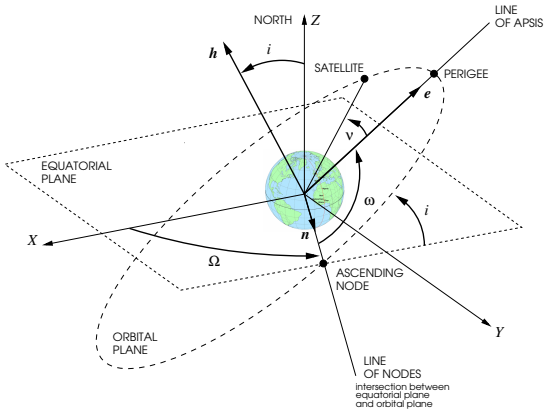

Figure 1: Classical Orbital Elements (COEs).

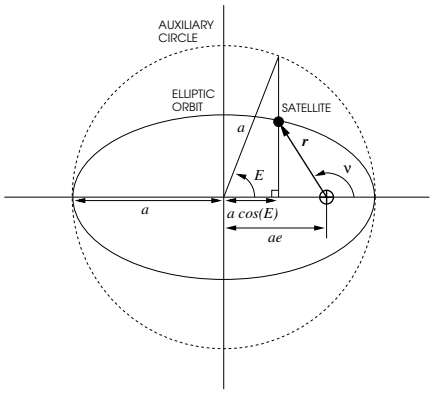

Figure 2: True and eccentric anomalies for elliptic motion.

(see, e.g., [1, 5, 8] for details)

$$a, \tag{11}$$

$$P_1 = e\sin(\omega + I\Omega), \tag{12}$$

$$P_2 = e\cos(\omega + I\Omega), \tag{13}$$

$$Q_1 = \tan(i/2)\sin\Omega, \tag{14}$$

$$Q_2 = \tan(i/2)\cos\Omega, \tag{15}$$

$$l = \Omega + \omega + M. \tag{16}$$

True retrograde equatorial orbits ($i = 180°$) cause problems because Q_1 and Q_2 are undefined. This problem is solved by introducing a retrograde factor I which is $+1$ for direct orbits and -1 for retrograde orbits. In this work, dealing with geostationary satellites, I is equal to $+1$ and the mean longitude net of the Greenwich Hour Angle $\Theta(t)$

$$l_\Theta = l - \Theta(t) \tag{17}$$

will be used instead of the mean longitude l given by equation (16). GEO satellite state representation in terms of equinoctial orbital elements will be denoted as follows

$$x_{EOE} = \begin{bmatrix} a & P_1 & P_2 & Q_1 & Q_2 & l_\Theta \end{bmatrix}^T. \tag{18}$$

The definitions of the EOEs are referenced to the equinoctial reference frame, which can be obtained from the ECI reference frame by a rotation through the angle Ω about the Z axis, followed by a rotation through the angle i about the new X axis (which points in the same direction as the node vector n pointing the ascending node), followed by a rotation through the angle $-I\Omega$ about the new Z axis (which points in the same direction as the h vector). In the equinoctial frame the elements P_1 and P_2 represent the projection of the eccentricity vector onto the Q and E directions, respectively (see Figure 3). The elements Q_1 and Q_2 represent the projection of the vector oriented in the direction of the ascending node with magnitude $\tan(i/2)$, onto the Q and E directions, respectively. Note that in the singular cases of circular (or equatorial) orbits, the vector P (or Q) becomes zero; the indetermination in the two components of each vector is thus not a problem.

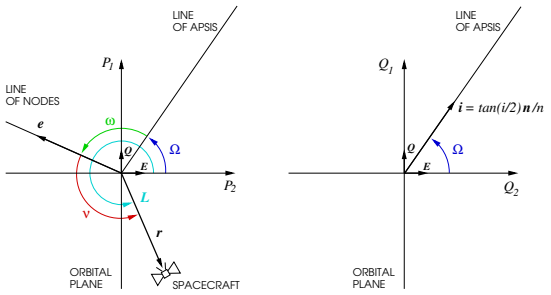

Figure 3: Eccentricity and inclination equinoctial components and true longitude.

2.4 Conversion formulae: COEs to Cartesian

The position coordinates in the orbital plane, centered in the Earth (Figure 2) are related to the COEs by the following equations

$$\begin{bmatrix} x_{orb} \\ y_{orb} \end{bmatrix} = \begin{bmatrix} a\cos(E) - ae \\ a\sin(E)\sqrt{1-e^2} \end{bmatrix}. \tag{19}$$

while the corresponding velocities can be computed as

$$\begin{bmatrix} v_{x,orb} \\ v_{y,orb} \end{bmatrix} = \begin{bmatrix} -\frac{a^2 n}{|r|}\sin(E) \\ \frac{a^2 n}{|r|}\sqrt{1-e^2}\cos(E) \end{bmatrix}, \tag{20}$$

with $|r| = \sqrt{x_{orb}^2 + y_{orb}^2} = \sqrt{r^T r}$. As depicted in Figure 1, the orthogonal basis RTN of the Gaussian coordinate system can be obtained from the orthogonal basis XYZ of the ECI frame by means of three successive rotations

$$\begin{bmatrix} x_{orb} \\ y_{orb} \\ 0 \end{bmatrix} = \mathscr{R}_{ZXZ}(x_{COE}) \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \tag{21}$$

with

$$\mathscr{R}_{ZXZ}(x_{COE}) = \mathscr{R}_Z(\omega)\mathscr{R}_X(i)\mathscr{R}_Z(\Omega) \tag{22}$$

where matrix

$$\mathscr{R}_Z(\Omega) = \begin{bmatrix} \cos\Omega & \sin\Omega & 0 \\ -\sin\Omega & \cos\Omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{23}$$

describes the first rotation around the Z axis of an angle Ω, matrix

$$\mathscr{R}_X(\Omega) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i & \sin i \\ 0 & -\sin i & \cos i \end{bmatrix} \tag{24}$$

describes the second rotation around the X of an angle i, matrix

$$\mathscr{R}_Z(\omega) = \begin{bmatrix} \cos(\omega) & \sin(\omega) & 0 \\ -\sin(\omega) & \cos(\omega) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{25}$$

describes the third rotation around the Z axis of an angle ω. Thanks to the orthonormal property of rotation matrices, equation (21) can be easily inverted, giving

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathscr{R}_Z^T(\Omega)\mathscr{R}_X^T(i)\mathscr{R}_Z^T(\omega) \begin{bmatrix} x_{orb} \\ y_{orb} \\ 0 \end{bmatrix}; \tag{26}$$

following the same reasoning, the Cartesian velocity vector can be expressed as

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \mathscr{R}_Z^T(\Omega)\mathscr{R}_X^T(i)\mathscr{R}_Z^T(\omega) \begin{bmatrix} v_{x,orb} \\ v_{y,orb} \\ 0 \end{bmatrix}. \tag{27}$$

Further details can be found, e.g., in [11]. Summarizing, it is possible to compute x_{COE}, given x_{ECI}, by first solving the scalar implicit equation (6) for E, and then the explicit vector equations (19)-(20), (26)-(27).

2.5 Conversion formulae: EOEs to Cartesian

The conversion formulae from EOEs to Cartesian coordinates in ECI are slightly more involved. The results are summarised here; for further details, the reader is referred to, e.g., [1, 5, 8].

The eccentric longitude K can be computed by solving the implicit equation

$$l_{theta} + \Theta(t) = K + P_1\cos(K) - P_2\sin(K). \tag{28}$$

The ECI coordinates are then given by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \rho \begin{bmatrix} (1+Q_2^2-Q_1^2)\cos(L) + 2Q_1Q_2\sin(L) \\ (1+Q_1^2-Q_2^2)\sin(L) + 2Q_1Q_2\cos(L) \\ 2Q_2\sin(L) - 2Q_1\cos(L) \end{bmatrix}, \tag{29}$$

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = n \begin{bmatrix} \frac{x}{|r|}\frac{d|r|}{dl} + \sigma\left[(1+Q_2^2-Q_1^2)\frac{dcos(L)}{dK} + 2Q_1Q_2\frac{dsin(L)}{dK}\right] \\ \frac{y}{|r|}\frac{d|r|}{dl} + \sigma\left[(1+Q_1^2-Q_2^2)\frac{dsin(L)}{dK} + 2Q_1Q_2\frac{dcos(L)}{dK}\right] \\ \frac{z}{|r|}\frac{d|r|}{dl} + 2\sigma\left[Q_2\frac{dsin(L)}{dK} - Q_1\frac{dcos(L)}{dK}\right] \end{bmatrix}, \tag{30}$$

where

$$\begin{aligned} |r| &= a(1 - P_1\sin(K) - P_2\cos(K)), \\ \rho &= \frac{|r|}{1+Q_1^2+Q_2^2}, \\ \sigma &= \frac{a}{1+Q_1^2+Q_2^2}, \\ \gamma &= 1 + \sqrt{1-P_1^2-P_2^2} \\ \sin(L) &= \frac{a}{\gamma|r|}\left[(\gamma - P_2^2)\sin(K) + P_1P_2\cos(K) - \gamma P_1\right] \\ \cos(L) &= \frac{a}{\gamma|r|}\left[(\gamma - P_1^2)\cos(K) + P_1P_2\sin(K) - \gamma P_2\right]. \end{aligned}$$

3 COEs and EOEs for simulation of orbit dynamics

When orbital control problems are studied, it is usually necessary to integrate the equations of motion of the satellite under the action of gravity (due to the Earth or any other celestial body), of the space environment and of the actuators' thrust. The usual approach, known as Cowell's method (see [3]), is to integrate the equations of motion in cartesian coordinates

$$\dot{r} = v \tag{31}$$

$$\dot{v} = a_g(r) + \frac{F}{m} \tag{32}$$

where a_g is the acceleration of gravity, F is the sum of all the other forces, and m is the satellite mass. applied by the actuators. First-cut models assume a point-mass model

$$a_g = -GMr/\|r\|^3, \tag{33}$$

while accurate simulations require more detailed models of the gravitational field, usually in the form of a series expansion (see, e.g., [12]). In both cases, the differential equations are strongly non-linear; therefore, despite the use of high-order integration algorithm, tight tolerances end up in a fairly high number of simulation steps per orbit.

If the satellite motion is described in terms of COEs or EOEs, it is easy to observe that the variability of the six orbit elements is much smaller than that of the Cartesian coordinates. In particular, it is well-known that in case of a point-mass gravity field with no other

applied forces, the first five parameters are constant, while the mean anomaly and the mean longitude increase linearly with time. All existing high-order integration methods have error bounds which depend on Taylor expansions of the state trajectory. One can then conjecture that if the COEs/EOEs are used as state variables, instead of the Cartesian vectors r and v, the state trajectories will be smoother, and therefore the integration algorithm will be able to estimate them with with a higher relative precision using much larger time steps, compared to the Cartesian coordinates case.
Recalling the definition of vector x_{ECI} in (4), letting $z = x_{COE}$ or $z = x_{EOE}$ depending on the choice for the new state variables and denoting by $g(\cdot)$ the transformation relating z and x, equations (31)-(32) can be written in compact form as

$$\dot{x} = f(x) \tag{34}$$
$$x = g(z). \tag{35}$$

If a state variable change from x to z is now performed, the following equations are obtained

$$\frac{\partial g(z)}{\partial z}\dot{z} = f(g(z)) \tag{36}$$

which can be solved for $\dot{z}$ provided that the new state variables z are uniquely defined

$$\dot{z} = \left(\frac{\partial g(z)}{\partial z}\right)^{-1} f(g(z)) \tag{37}$$
$$x = g(z). \tag{38}$$

The Jacobian for g_{COE} is generically well defined and becomes singular only in the case of a circular and/or equatorial orbit. In this case the EOEs are needed, as the Jacobian for g_{EOE} is well defined in this case.
The model (37)-(38), which is now in standard state-space form, has two very important features:

- the right-hand side of (37) is much less variable than the right-hand side of (34), so it will be easier to integrate the equations with a higher accuracy;
- in case an accurate model of the gravity field is used, it is not necessary to reformulate it in terms of the COEs/EOEs, as the right-hand side of (37) uses the compound function $f(g(z))$.

Remark 1 *The accurate computation of long-term solutions for dynamical problems associated with pure orbital motion has been a subject of extensive research for decades. In particular, the so-called class of symplectic integration methods (see, e.g., [4] and the references therein) provides an effective and reliable solution to the problem. In the framework of the present study, however, the aim is to improve accuracy in the computation of orbital motion while retaining the advantages associated with the use of a general-purpose object-oriented modelling environment, in which not only orbital dynamics can be simulated, but also the coupled attitude motion, as well as the associated mathematical models of sensors, actuators and controllers for orbital and attitude control. This more general framework requires the use of general-purpose integration algorithms for ODEs/DAEs.*

4 Modelica implementation

The concepts outlined in Section 3 are easily implemented with the Modelica language. The starting point is the Body model of the standard Modelica.Mechanics.MultiBody library [9]: this is a 6 degrees-of-freedom model of a rigid body, which can be connected to other components to form a multibody system model. The original model has six degrees of freedom, corresponding to 12 state variables: the three cartesian coordinates and the three velocity components of the center of mass, plus three suitable variables describing the body orientation and the three components of the angular velocity vector. Assuming that the gravitational field is applied exactly at the center of mass (the gravity gradient effect is computed in a separate model and thus not included here), the translational and rotational equations are completely decoupled, so it is possible to focus on the former ones, leaving the latter ones untouched.
First of all, the equations to compute the gravity acceleration *as a function of the cartesian coordinates* using accurate field models are added by inheritance to the standard World model of the MultiBody library, which only offers the most basic options of no gravity, constant gravity and point mass gravity (see [7, 10]). Then, the standard Body model must be enhanced by:

1. adding the COEs a, e, i, ω, Ω, M or the EOEs a, P_1, P_2, Q_1, Q_2, l_Θ as new model variables;
2. adding the equations relating COEs/EOEs to the cartesian coordinates;
3. switching the stateSelect attribute for the r and v vectors of the Body model to StateSelect.avoid, and for the COEs/EOEs to StateSelect.prefer.

The Modelica compiler tool will then perform the transformation from (34)-(35) to (37)-(38) automatically, using symbolic manipulation algorithms.

A first implementation option is to extend the Body model by inheritance, adding the above-mentioned features, and thus deriving two enhanced models BodyKepler and BodyEquinoctial; this approach is documented in [2].

A second option is to put the additional variables and equations in a separate model with a multibody flange interface, and then connect it to the unmodified Body model within a wrapper model that also sets the preferred state variables. This option perfectly fits the architecture of the Spacecraft Dynamics library, where such a structure was already used in order to include the models of the interaction of the satellite with the space environment: gravity gradient torque, aerodynamic drag, solar radiation, etc. (see [10], Fig. 3). In fact, the library described in [10] already contained a similar model to compute the orbital parmeters; that model, however, contained explicit inverse conversion formulae (from cartesian coordinates to COEs), and was designed to be used with cartesian coordinates as states. Since either the COEs or the EOEs can be used, the wrapper model must actually contain two conditionally declared, mutually exclusive models (one for each choice of coordinates), which are both connected to the standard Body model; a flag in the wrapper model decides which of the two will actually be activated in the simulation model.

The Modelica code defining the new models is very compact and easy to check, which is an important feature to ensure the correctness of the resulting model. As already noted, the accurate models of the gravity field, previously implemented in [7, 10], can still use the Cartesian coordinates as inputs, and are thus left unchanged.

As to the computational efficiency, the workload at each time step is increased, compared to the standard ECI formulation, by the conversion formulae, the Jacobian computation and the solution of the linear system (37). However, as will be demonstrated in the next section, this additional overhead is more than compensated by the fact that the differential equations are much easier to integrate in the new state variables, resulting in a faster simulation time and in a much tighter accuracy.

5 Simulation examples

In this Section, the results obtained in comparing the accuracy obtained by simulating the orbit dynamics for two Low Earth orbiting (LEO) spacecraft and a GEO one will be presented. As previously mentioned, for the purpose of the present study we focus on the simulation of the unperturbed dynamics, i.e., only the gravitational acceleration computed from a point-mass model for the Earth is considered. In this case, the orbit is an ellipse (closed curve), having well-defined features. Therefore, this assumption allows us to introduce two simple criteria in order to evaluate the accuracy of the performed simulations, namely:

- The period of an unperturbed elliptical orbit can be computed a priori and is given by $T = 2\pi\sqrt{\frac{a^3}{\mu}}$, so a first measure of simulation accuracy can be given by the precision with which the orbit actually closes during the simulation. To this purpose, the following stopping criterion has been defined for the simulation: the integration is stopped when the position vector crosses a plane orthogonal to the initial velocity and passing through the initial position. Then, the final time is compared with the orbit period and the final position is compared with the initial one.

- Furthermore, for an unperturbed orbit the angular momentum $h = r \times v$ should remain constant, so a second measure of accuracy for the simulation is given by the relative error in the value of h, i.e., the quantity

$$e_h = \frac{\|h - h(0)\|}{\|h(0)\|}. \tag{39}$$

The considered orbits have been simulated using the Dymola tool, using Cartesian and Keplerian/Equinoctial coordinates, in order to evaluate the above-defined precision indicators. The DASSL integration algorithm has been used, with the smallest feasible relative tolerance 10^{-12}. The RADAU algorithm has also been tried with the same relative tolerance, yielding similar results which are not reported here for the sake of conciseness.

5.1 A near-circular, LEO orbit

The first considered orbit is a LEO, near circular one (see Figure 4), characterised by the following initial

state, in Cartesian coordinates:

$$r(0) = \begin{bmatrix} 6828.140 \times 10^3 \\ 0 \\ 0 \end{bmatrix},$$

$$v(0) = \begin{bmatrix} 0 \\ 5.40258602956241 \times 10^3 \\ 5.40258602956241 \times 10^3 \end{bmatrix}$$

The results obtained in the comparison of Cartesian and Keplerian coordinates are summarised in Table 1. As can be seen from the Table, the precision achieved in the actual closure of the orbit improves significantly when using Keplerian coordinates as states: the simulated period is very close to the actual one and both the period error and the position error are significantly smaller.

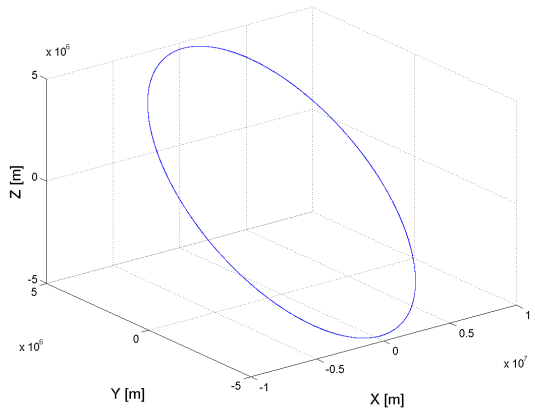

Figure 4: The considered LEO, near circular orbit.

Similarly, in Figure 5 the time histories of the relative error on the value of the orbital angular momentum are illustrated, for a simulation of about one day: the results are clearly very satisfactory in both cases, however while in the case of Cartesian states the relative error is significantly larger than machine precision and is slowly increasing, in the case of Keplerian states the relative error is much smaller and appears to be more stable as a function of time (see also the mean value of the relative angular momentum error, given in Table 1). Finally, note that the use of Keplerian parameters also gives significant benefits in terms of simulation efficiency, as can be seen from the last column of Table 1.

5.2 A highly elliptical, LEO orbit

The second considered orbit is again a LEO one, but it is characterised by a high value of the eccentricity

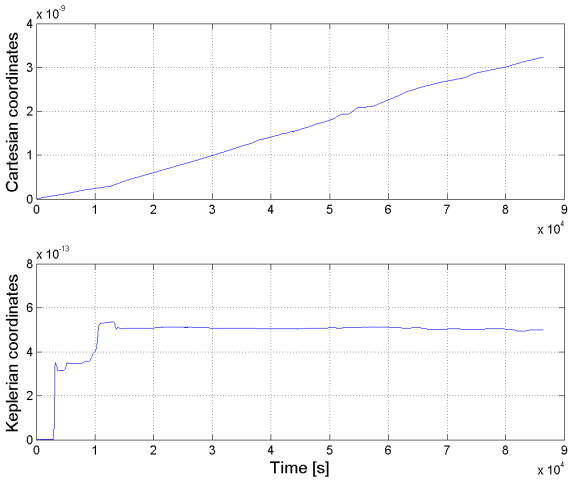

Figure 5: Relative errors on the orbit angular momentum - near circular orbit: Cartesian (top) and Keplerian (bottom) coordinates.

(see Figure 6, where it is also compared with the circular orbit considered in the previous case) and by the following initial state, in Cartesian coordinates:

$$r(0) = \begin{bmatrix} 6828.140 \times 10^3 \\ 0 \\ 0 \end{bmatrix},$$

$$v(0) = \begin{bmatrix} 0 \\ 5.40258602956241 \times 10^3 \\ 7.29349113990925 \times 10^3 \end{bmatrix}$$

As in the previous case, Table 2 shows the precision achieved in the actual closure of the orbit: as can be seen, the errors on the simulated period are of the same order of magnitude for both choices of state variables. The position errors, on the other hand are significantly smaller when simulating the orbital motion using Keplerian rather than Cartesian states.

Similarly, in Figure 7 the time histories of the relative error on the value of the orbital angular momentum are illustrated, for a simulation of about one day. In this case, the results show that using Cartesian states the relative error is again significantly larger than machine precision and is slowly increasing, while using Keplerian states the relative error is of the order of machine precision.

Finally, the gain in terms of simulation efficiency can be verified from the last column of Table 2.

Table 1: Orbit closure errors, relative angular momentum error and number of steps using Cartesian and Keplerian coordinates - near circular orbit.

States	ΔT [s]	$\|\Delta r\|$ [m]	Mean e_h	Number of steps
Cartesian	-1.00332×10^{-6}	1.69711×10^{-3}	1.5373×10^{-9}	959
Keplerian	2.38369×10^{-8}	2.17863×10^{-5}	4.7528×10^{-13}	376

Table 2: Orbit closure errors, relative angular momentum error and number of steps using Cartesian and Keplerian coordinates - highly elliptical orbit.

States	ΔT [s]	$\|\Delta r\|$ [m]	Mean e_h	Number of steps
Cartesian	-1.17226×10^{-5}	4.39241×10^{-3}	1.2927×10^{-10}	3650
Keplerian	1.48665×10^{-5}	2.67799×10^{-7}	2.5223×10^{-16}	1120

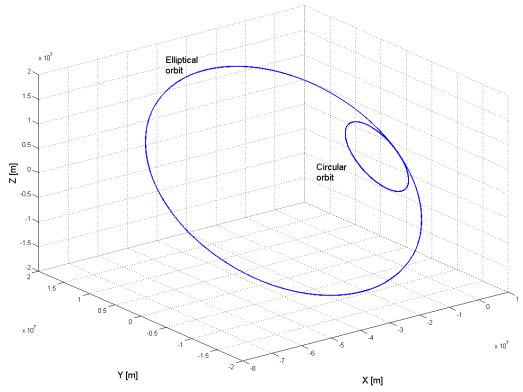

Figure 6: The considered LEO, highly elliptical orbit, compared with the circular one considered in Section 5.1.

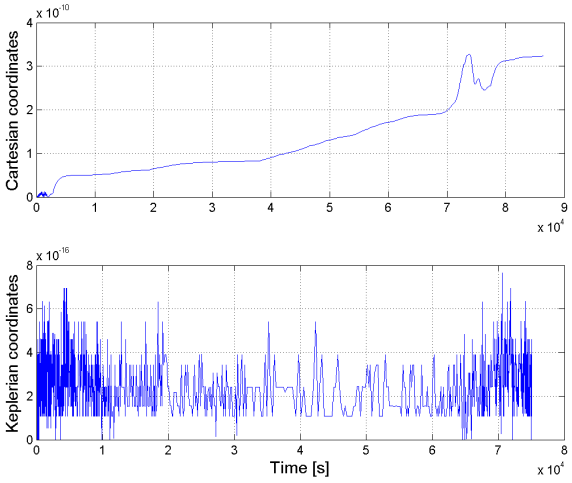

Figure 7: Relative errors on the orbit angular momentum - highly elliptical orbit: Cartesian (top) and Keplerian (bottom) coordinates.

5.3 A GEO orbit

The last considered orbit is a GEO one, characterised by the following initial state, in Cartesian coordinates:

$$r(0) = \begin{bmatrix} 4.21641 \times 10^7 \\ 0 \\ 0 \end{bmatrix},$$

$$v(0) = \begin{bmatrix} 0 \\ 3074.66 \\ 0 \end{bmatrix}$$

Table 3 shows the accuracy improvement achieved when simulating the orbital motion using Equinoctial rather than Cartesian states. As in the previous case, also for the simulation of GEO orbits it appears from the inspection of the time histories of the relative error on the orbital angular momentum (depicted in Figure 8) that in the case of Cartesian states the relative error is slowly increasing over time, while in the case of Equinoctial states the relative error appears to be more stable (see also Table 3).
Finally, the advantages provided by the use of Equinoctial parameters in terms of simulation efficiency are confirmed by the data provided in the last column of Table 3.

6 Concluding remarks

A method for the accurate simulation of satellite orbit dynamics on the basis of the Modelica MultiBody library has been presented. The proposed approach is based on the use of Keplerian and Equinoctial parame-

Table 3: Orbit closure errors, relative angular momentum error and number of steps using Cartesian and Equinoctial coordinates - GEO orbit.

States	ΔT [s]	$\|\Delta r\|$ [m]	Mean e_h	Number of steps
Cartesian	-2.79186×10^{-5}	1.88208×10^{-2}	1.0323×10^{-10}	793
Equinoctial	-2.92057×10^{-8}	8.91065×10^{-5}	6.8574×10^{-16}	20

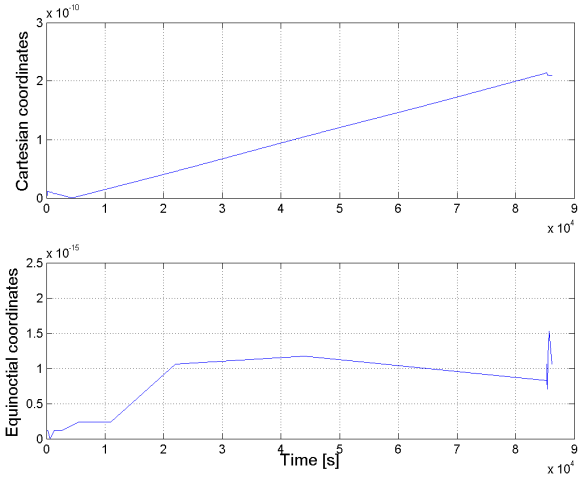

Figure 8: Relative errors on the orbit angular momentum - GEO orbit: Cartesian (top) and Equinoctial (bottom) coordinates.

ters instead of Cartesian coordinates as state variables in the spacecraft model. This is achieved by adding to the standard Body model the equations for the transformation from Keplerian and Equinoctial parameters to Cartesian coordinates and exploiting automatic differentiation. The resulting model ensures a significant improvement in numerical accuracy and a reduction in the overall simulation time, while keeping the same interface and multibody structure of the standard component. Simulation results with a point-mass gravity field show the good performance of the proposed approach. The validation with higher order gravity field models is currently being performed.

References

[1] R.A. Broucke and P.J. Cefola. On the equinoctial orbit elements. *Celestial Mechanics and Dynamical Astronomy*, 5(3):303–310, 1972.

[2] F. Casella and M. Lovera. High accuracy simulation of orbit dynamics: an object-oriented approach. In *Proceedings of the 6th EUROSIM Congress, Lubiana, Slovenia*, 2007.

[3] V. Chobotov. *Orbital Mechanics*. AIAA Education Series, Second edition, 1996.

[4] V.V. Emel'yanenko. A method of symplectic integrations with adaptive time-steps for individual Hamiltonians in the planetary N-body problem. *Celestial Mechanics and Dynamical Astronomy*, 98(3):191–202, 2007.

[5] D. Losa. *High vs low thrust station keeping maneuver planning for geostationary satellites*. PhD thesis, Ecole Nationale Superieure des Mines de Paris, 2007.

[6] M. Lovera. Object-oriented modelling of spacecraft attitude and orbit dynamics. In *54th International Astronautical Congress, Bremen, Germany*, 2003.

[7] M. Lovera. Control-oriented modelling and simulation of spacecraft attitude and orbit dynamics. *Journal of Mathematical and Computer Modelling of Dynamical Systems, Special issue on Modular Physical Modelling*, 12(1):73–88, 2006.

[8] O. Montenbruck and E. Gill. *Satellite orbits: models, methods, applications*. Springer, 2000.

[9] M. Otter, H. Elmqvist, and S. E. Mattsson. The new Modelica multibody library. In *Proceedings of the 3nd International Modelica Conference, Linköping, Sweden*, 2003.

[10] T. Pulecchi, F. Casella, and M. Lovera. A Modelica library for Space Flight Dynamics. In *Proceedings of the 5th International Modelica Conference, Vienna, Austria*, 2006.

[11] M. Sidi. *Spacecraft dynamics and control*. Cambridge University Press, 1997.

[12] J. Wertz. *Spacecraft attitude determination and control*. D. Reidel Publishing Company, 1978.