Abstract

This paper describes the implementation, testing and validation of an Attitude Dynamics Module as part of a GNC analysis and design software tool for the Control Systems Division of the Directorate of Technical and Quality Management at the European Space Agency (ESA). The main contribution of this thesis is the propagation of attitude, given an initial attitude state, in different formats. Furthermore, the output is formatted to present the analysis and visualisation of the attitude in time. The second contribution is the extension of an analysis module with routines for the representation of trajectories in different coordinate systems and time frames. The result is a tool that can propagate position and attitude of space systems, and produce output in several reporting forms to the user’s convenience.

I Introduction

Space simulation software tools aim to recreate, as close as possible to reality, real phenomena, using a set of mathematical models to do so. Through these models almost real space environments can be simulated. They are most commonly used during early phases of a project, permitting an intense study of all the stages involved on a space mission, such as launch, loitering, re-entry, interplanetary amongst others.

The Attitude Dynamics Module was created as a response to the need to know the orientation of a vehicle during its mission in space mission analysis and design software tools.

Attitude control is critical for the positioning of satellites, for the goal is to keep them oriented in a pre-determined and specified direction. Moreover, orbit control is very important when it comes to other systems of a spacecraft, for example, getting the right position of the solar arrays, radiators or antennas, apply the thrusters in the right direction or controlling the vehicle’s orientation during re-entry. The simulation software should also be able to present the attitude in different formats to the user, such as different coordinate systems or time frames, and allow the visualisation of the results, so that a correct mission plan and attitude control is prepared.

The present work develops two modules to this end: the Attitude Dynamics Module (ADM) and further develops an existing Analysis Module (AM).

The ADM allows the specification of an initial attitude in different representations, and its propagation in time. The ADM produces an output formatted to present the analysis and visualisation of the attitude’s variation in time. For further analysis, a module that can analyse the outputs and transform them as desired by the user is necessary. The
Analysis Module permits the plotting of
the output in several forms, for an ade-
quate reporting, being capable of per-
forming transformations between coordi-
nate systems and time frames.

II Existing mission analy-
sis tools
In the context of mission analysis, there
are several tools that allow the analy-
sis of space missions. These wide range
of astrodynamics tools may be an open
source or commercial software, or have
both types of versions. In the simulations
tools one may find: Dance of the Plan-
ets, FreeFlyer, GMAT, Orbimat, STK
or Starmad, among many others. While
Dance of the Planets is used to simu-
late celestial events, being a top simu-
lator for astronomy, GMAT or Orbimat
are more used to analyse and calculate
mission parameters and trajectory opti-
misations. Orbits, trajectories, attitude
and GNC are often studied using these
type of software.

III Requirements of the
Attitude and Analysis
Modules
III.1 Attitude Dynamics Mod-
ule
The main goal of the ADM is to control
the spacecraft’s attitude, by orienting it
in a specified and predetermined direc-
tion. But before doing so, the module
needs to be able to propagate an initial
attitude. Therefore, the ADM will re-
ceive the initial attitude in several atti-
dude formats (Euler angles - in three dif-
f erent sequences - or quaternions), as well
as the inertia matrix and body rates and
will use mathematical models to propa-
gate it into a desired final attitude for the
spacecraft mission time. To accomplish
these requirements the ADM shall have
a GUI where the input of all the param-
eters is made, and an output where the
generation of text reports and 2D plots is
used to analyse the data after the propa-
gation.

III.2 Analysis Module
One of the main objectives of this mod-
ule is to be able to perform several con-
versions between different coordinate sys-
tems and time frames in order to be able
to generate reports or 2D and 3D plots of
the selected data. The time scales shall
contain the following options: Grego-
rian UTC, Julian UTC, YYDDD, Julian
date, Mission elapsed time, Modified Ju-
lian Date, Julian Ephemeris Date, Grego-
rian: TDT, TDB, TAI, GPS; Julian
TDB, Earth canonical time. It shall also
be possible to analyse the position, veloc-
ity and acceleration as well as the Kep-
erian, Delaunay, Equinoctial and Spher-
ical elements of a given object in the fol-
lowing coordinate systems: ICRF, Mean
of Date and True of Date, which need
to be implemented in addition to the al-
ready present systems.

III.3 ADM and AM require-
ments with the code
To properly be integrated and tested
with a simulation software tool, both
modules must comply with the software’s
code requirements. These include: their
code must be compatible with Qt li-
braries and Eigen libraries; they shall
have the ability to be extended by adding
classes and they shall have a consistent
user interface that shall be properly inte-
grated in the simulation tool.

IV The Attitude Dynam-
ics Module’s Graphical
User Interface (GUI)
The Attitude Dynamics Module interacts
with the user through a Graphical User
Interface, or GUI. In this GUI, the user
inputs the initial conditions for attitude
propagation. After being read and saved, the input is sent to a module where the mathematics are made to perform the propagation. The outputs of the ADM will later be sent to the Analysis Module where the data may be analysed with reports or plots of the propagation’s results.

IV.1 Initial Attitude’s GUI

Figure 1: Screen shot of the Initial Attitude GUI, with the initial attitude given in Euler angles and body rates

In figure 1 it is shown the initial attitude’s GUI. The input for attitude may be expressed in Euler angles, in the 123, 321 or 313 sequences or in Quaternions ESA or Quaternions JPL (the relation between the two definitions of quaternions is the following: \(q_{1\text{JPL}} = q_{2\text{ESA}}, q_{2\text{JPL}} = q_{3\text{ESA}}, q_{3\text{JPL}} = q_{4\text{ESA}}, q_{4\text{JPL}} = q_{1\text{ESA}}\), where for ESA definition the real part of the quaternion is \(q_{1\text{ESA}}\), while in JPL’s definition is \(q_{4\text{JPL}}\)).

To correctly input attitude properties the following fields in the GUI must be filled:

- **Starting epoch:** this allows the user to specify the start epoch to start attitude propagation;
- **Central body:** choice of the satellite to propagate, as the origin of the coordinate system for the attitude propagation;
- **Coordinate system:** in this first version only the inertial coordinate system is available, and the origin of this system needs to be specific;
- **Type:** this box allows the selection of the initial attitude, as in Euler or Quaternions. For Euler angles there are three sequences available (123, 321, 313), and for quaternions two possibilities: Quaternions ESA or Quaternions JPL.
- **Initial Attitude input:** for the user to input the angles and body rates (either in radians or degrees), or the initial quaternions and quaternion rates.

IV.2 Propagation’s GUI

Figure 2: Screen shot of the Attitude Propagation GUI

The inputs needed for the attitude propagation are:

- **Ending epoch:** this box defines the end of the attitude propagation;
- **Propagator type:** for now, attitude is only propagated in quaternions. This may be expanded later to propagate in terms of Euler angles;
Integrator type: for now only Runge-Kutta 3-4 is available. Other integrators may be added in later versions of ADM;

Integration step: defined the integration step to be used by the integrator to propagate attitude.

The fields Tolerance ABS and Tolerance REL are not used with Runge-Kutta 3-4 integrator, due to the fact that this integrator has fixed step size.

V The Analysis Module’s Graphical User Interface (GUI)

Through the Analysis Module the user selects the coordinate system and the time frame in which he wants to analyse the object’s results for the space mission. The type of coordinate system chosen depends on the mission requirements for the accuracy. The following coordinate systems are present in the analysis module: ICRF, Mean of Date and True of Date. Time is also a very sensitive subject, and the frame in which it is outputted also depends on the accuracy needed for the mission analysis.

The Analysis Module’s GUI is divided in two parts. In the first part the user shall select the trajectory plan and the start and end epochs of the mission. Both time and time step can be set as default or the user may input the time interval and time step. The default option uploads the time intervals and time step specified in the scenario.

Afterwards, the parameters to be analysed shall be selected by clicking the “Parameters” tab, figure 3. The parameters shall be selected on the left part of the GUI, and moved to the right part by clicking the “>” button, and can be removed by clicking the “<<” button.

VII Verification and Validation

VI.1 Introduction

In order to validate a software, several test cases must be made. After it is validated, the software must run without any errors and the physics behind the module must give the correct results. Both modules must comply with the requirements specified in the Software Requirements Document of the Attitude Dynamics Module [1], and the Software Requirements Document of the Analysis Module [2].

VI.2 Attitude Dynamics Module

VI.2.1 Project Requirements

The ADM must be verified in all its aspects before it can be considered validated. Therefore, the verification plan must be a comprehensive set of tests which checks all the functions of the software until there can be no more errors.

These tests must ensure that the all the data needed for the input of initial attitude and attitude propagation are inside the GUI. The mathematics behind the attitude module shall be verified and validated. The tests include the verification and validation of the conver-
sion between the three Euler sequences, and between the Euler sequences and quaternions, and the correct integration an propagation of attitude.

VI.2.2 Tests Description
To completely validate the module, six sets of tests were performed. The first test aims to verify if all the parameters needed for attitude propagation are inside the GUI. After that, the transformations between the several parameters, such as Euler angles, quaternions, body rates and quaternion rates are tested in order to assure the conversions are made correctly. Finally, the propagation is then tested and the output is verified.

The first test is verified through an inspection process, the conversions and propagation are made by testing, comparing the results with different software.

VI.2.3 Test Results

Attitude GUI
Through inspection of the GUI, it was verified that all the elements were inside the attitude GUI, for both the input of the initial attitude and the parameters for attitude propagation. These include: the initial attitude in Euler angles - in three different sequences, the quaternions - in two different definitions, the start and end epoch, the time step, the propagator and integrator types and the inertia matrix.

Conversion between the three Euler sequences
In this first version of the ADM, three of the twelve different sequences for Euler angles are tested - 123, 321 and 313 sequences. This test is to validate the correct conversion between the three sequences inside the initial attitude’s GUI, and it is valid if the results match Matlab’s results for the same initial conditions. To this end, two scenarios were created: the first one is a general case and in the second one a singularity will be present.

For the general case, it was shown that the results were identical to Matlab’s results, therefore validating this routine. Both had the same number of significant digits which result in a null error between the two software.

For the second test, and as expected the ADM did not perform any conversion, giving a “not a number” message. This is due to the fact that when there is a singularity present the ADM is not able to perform the conversion neither to another sequence nor to quaternions.

Conversion between the Euler angles and quaternions
In this test, the conversion between Euler angles and quaternions are tested. The results will be compared with GMAT software, and the two scenarios will be tested: the general case with the input given in the 321 sequence and the second case will have the input given in the 313 sequence. This test will not consider the input of a singularity, because, as it was shown in the previous test, with a singularity the output of the conversion is “not a number”.

The output is shown in the two configurations of quaternions - Quaternions ESA and Quaternions JPL - and the maximum relative error is 0.000106378% for the first scenario, which is minor than the 0.05% of limit imposed by ESA, for this project.

Conversion between the body rates and quaternion rates
This test compares ADM’s results with Matlab’s, and aims to validate the correct conversion between body rates and quaternion rates. Once again, two scenarios are considered: the first one is the general case, with the input given in the 321 sequence and all body rates have values different from zero, and in the second case, the input is given in the 313 sequence and two body rates are set to
From all the tests made to validate this routine, the maximum relative error encountered was 0.0002606%. This error complies with ESA requirement, and for this reason this routine is validated.

**Visualisation of the results**

The visualisation of the results of the propagation is made with the Analysis Module. To this end, a link between the two modules was performed, being possible to plot and generate reports with the results of the propagation using all the capabilities of the AM.

The results of the propagation may be plotted against any other variable of the space scenario's participant.

In figure 4, it is shown an example, where the first rotation angle, of a 321 sequence, was plotted against the Z position of the vehicle.

The initial Euler angles are $\phi = 23^\circ$, $\theta = 34^\circ$ and $\Psi = 45^\circ$, and as it is shown by figure 5, they remain constant throughout all the mission duration.

The other test aims to validate the general case, where all the values are different from zero, having the satellite an initial attitude of 10° on each axis, and angular velocity of 1 degree per second, also on each axis. These conditions lead to a vehicle with a tumbling movement where not all the axis perform a complete revolution. The result of this propagation is shown on figure 6.
VI.3 Analysis Module

VI.3.1 Project Requirements

In the same way as the Attitude Dynamics Module, the Analysis Module shall be tested in all its aspects: the software must run without any errors and the mathematics behind it must give the correct results. The verification and validation process shall ensure that the requirements listed in the Software Requirements Document of the Analysis Module [2] are complied. The GUI of this module shall be verified, the conversions between the different coordinate systems and time frames shall give the correct results and the output shall be expressed in the form of reports or plots.

VI.3.2 Tests Description

The first test will be made by inspection to the analysis’s GUI, in order to assure that all the elements are inside. After that, the mathematics will be tested for the distance and angular conversions, the coordinate systems transformations and time scales conversions. Finally, the output will be analysed by inspection of the results.

VI.3.3 Test Results

Verification of the GUI

By inspection of the GUI, it was shown that all the elements are inside the analysis’s GUI. These include: all the time frames, the State Vector, Keplerian, Delaunay, Spherical and Equinoctial elements with all the coordinate systems and the different distance and angular units.

Conversion of distance units

The conversion between kilometers, meters, astronomical units and nautical miles inside the GUI were verified and validated in this test.

Conversion of angular units

In the same way, the conversion between degrees and radians inside the GUI were verified and validated in this test.

Conversion between coordinate systems

Three new coordinate systems were implemented on the analysis module: ICRF, Mean of Date and True of Date. This test validates the correct conversion between the several coordinate systems, comparing the results with a Matlab routine (for the validation of ICRF) and Orbimat. The results for the position vector are display in table 1.

Table 1: Relative and Absolute errors for the position vector of the coordinate systems implemented

<table>
<thead>
<tr>
<th>System</th>
<th>Rel. Error [%]</th>
<th>Abs. Error [km]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICRF</td>
<td>5.053E − 13</td>
<td>6.26646E − 12</td>
</tr>
<tr>
<td>Mean of Date</td>
<td>7.898E − 06</td>
<td>0.00063971</td>
</tr>
<tr>
<td>True of Date</td>
<td>3.738E − 06</td>
<td>0.000302756</td>
</tr>
</tbody>
</table>

And the results for the velocity vector are displayed on table 2.

Table 2: Relative and Absolute errors for the velocity vector of the coordinate systems implemented

<table>
<thead>
<tr>
<th>System</th>
<th>Rel. Error [%]</th>
<th>Abs. Error [km]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICRF</td>
<td>1.9228E − 09</td>
<td>2.2602E − 10</td>
</tr>
<tr>
<td>Mean of Date</td>
<td>9.37211E − 07</td>
<td>6.57451E − 08</td>
</tr>
<tr>
<td>True of Date</td>
<td>9.41074E − 07</td>
<td>6.60161E − 08</td>
</tr>
</tbody>
</table>

The small errors encountered permit the validation of these routines.

Conversion of time frames

The conversion between the different time scales will be validated using an ESA internal software tool. To this end, a scenario was created with the start epoch given as: 23/03/2011 11:28:17 TDB and the ending epoch as: 24/03/2011 11:28:17 TDB, with a time step of 60 seconds. For the Gregorian dates and Julian UTC the results are the same, for the same number of significant
digits and therefore the error is zero. For the Earth canonical time, Modified Julian Date, Julian Ephemeris Date, Julian Date and YYDDD, the digits needed to express the date influence the error. For these dates, the relative error and the absolute error are shown in table 3.

Table 3: Relative and Absolute errors for the conversion between the time formats

<table>
<thead>
<tr>
<th>Time frame</th>
<th>Rel. Error</th>
<th>Abs. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earth Canonical Time</td>
<td>0.4962855909%</td>
<td>0.000469004</td>
</tr>
<tr>
<td>Modified Julian Date</td>
<td>3.8674E−09%</td>
<td>0.000125626</td>
</tr>
<tr>
<td>Julian Ephemeris Date</td>
<td>4.07228E−10%</td>
<td>1.00001E−05</td>
</tr>
<tr>
<td>Julian Date</td>
<td>8.7517E−11%</td>
<td>2.14996E−06</td>
</tr>
<tr>
<td>YYDDD</td>
<td>6.912128E−06%</td>
<td>0.000766041</td>
</tr>
</tbody>
</table>

The relatively big error in the Earth canonical time is due to the fact that ESA software presents the results with only four significant digits, against the ten digits presented by the Analysis Module. If the same number of significant digits are to be consider, then this error drops to zero.

Considering the low values of errors encounter in the remaining dates, it is possible to conclude that these conversions are made correctly, and therefore these routines are validated.

**Data output**

One of the major assets of the analysis module is its ability to generate plots and 2D and 3D plots. This feature will be tested by inspection. To that end, the generation of a report, and 2D and 3D plots was requested for the time, expressed in minutes from epoch, the x position and the z position. The results are shown in figures 7, 8 and 9.

As a conclusion, the analysis module is able to generate reports, 2D and 3D plots of any data selected on the “Parameters” tab. Furthermore, the analysis capabilities were increased, being now pos-
sible to select only a small part of the table to generate the graphic; the plots can be easily manipulated and edited; several graphics can be created simultaneously; there are more types of plotting options: Line, Scatter, Line+Scatter, Areas, Columns, Histograms, amongst many other new possibilities. With this new version of the analysis module there is also a very complete range of analysis possibilities to analyse the generated data, such as differentiate and integrate functions, interpolations, Fit functions like Gaussian, Exponential, Linear and Polynomial and Uncertainty Propagation analysis.

![Figure 10: Generation of Histograms for Altitude and X velocity](image)

![Figure 11: Generation of 2D plots for Time vs Altitude and Time vs X velocity](image)

**VII Conclusions**

Attitude determination and control is one the most important modules inside a GNC software tool. Without attitude the orientation of the satellite is not known, which has major implications on the success of the mission.

The Attitude Dynamics Module was developed from requirements phase to the verification and validation process, in order to overcome the lack of an attitude determination module inside certain space mission and design software tools.

As a conclusion, this module is able to receive an initial attitude, convert it to other Euler sequences and quaternions, and propagate it. The output is linked to the analysis module in order to allow a complete analysis of the propagation results.

The C++ functions used to implement this module were completely validated and the results comply with the requirements specified in the Software Requirements Document: the code can be extended by adding classes, there is a consistent user interface, the code is written using Qt and Eigen libraries and it was assured that there were no double routines inside the module and ADM can run in all the operative systems without the need of re-writing the code. Furthermore, the functions to convert between the three Euler sequences, between Euler and quaternions and between body rates and quaternion rates present relative errors below the limit imposed by ESA for this project (0.05%).

The Attitude Dynamics Module was completely validated when the propagation was validated with an ESA AOCS tool. It was shown that the propagation is held correctly.

The Attitude Dynamics Module was tested inside an ESA internal software tool, which was used as a test bench for this module. Although the module’s code style had to comply with the software’s, measures had to be taken to keep the module as independent as possible so it can be easily extracted. To accomplish this, all the transformations and propagation equations are independent C++ functions.

The Analysis Module was ex-
tended in order to accommodate more coordinate systems and time frames transformations. This module is important when it comes to analyse the data received from other modules inside a space simulation software tool. With these new functionalities the Analysis Module is able to produce text reports and plots of the Keplerian, Delaunay, Equinoctial and Spherical elements in the new coordinate systems: ICRF, Mean of Date and True of Date. New time scales were also implemented: Gregorian UTC, Gregorian TDT, Gregorian TDB, Gregorian TAI, Gregorian GPS, Julian Date, Julian TDB, Julian UTC, Julian Ephemeris Time, Modified Julian Date, Mission Elapsed Time, Earth canonical time and YYDDD. While expanding the Analysis Module, several code and transformation bugs were detected. This led to a first phase of bug corrections and validation prior to the coding of the new transformation routines. All current routines are now validated for correct representation. The analysis module is now part of an ESA internal software tool, and its expansion was a major contribution to the upgrade of the software.

References

