Spacecraft Orbital Computations Kit (SpOCK) User Guide

Sergio De Florio
sergio.deflorio@yahoo.com
Version 1

Contents

1 About SpOCK

The Spacecraft Orbital Computations Kit (SpOCK) is an open source tool for spacecraft mission analysis and simulation. SpOCK allows the simulation of spacecrafts' hardware, orbital and attitude dynamics, and the computations of mission events (ground station contacts, payload data-takes and eclipses). The ambition of the author is to develop an operational tool (mission analysis, flight dynamics and operations) which includes only few but accurate simulations features and which can be easily further developed and customized. For this reason the software is developed with a (hopefully) clear structure with elements which can be easily read and reproduced and without a graphical user interface.

Design

SpOCK is written in C++ language. The software has been built on the main requirements of modularity, customizability and use of already existing reliable and validated external libraries whenever possible. Among the huge number of features intrinsic to C++, class inheritance has been used extensively. This allows code reusability and readability which are fundamental features for a code which is devised to be cusomizable.

Core libraries for all computations are the Eigen and Boost libraries. Eigen is a fast and reliable C++ template library for linear algebra, matrices, vectors, numerical solvers, and related algorithms. Boost is a set of libraries for C++ that provide support for tasks and structures. Specifically, odeint which is a Boost library for solving initial value problems of ordinary differential equations, has been selected for the numerical integration of orbit and attitude dynamics.

The NASA SPICE library has been used for computation using planets and Sun ephemerides, for Earth-centered coordinate systems transformations and time systems transformations.

For the atmospheric models JB2008 and NRLMSISE-00 as well as for the Earth's magnetic fields models IGRF13 and WMM2020, the official Fortran libraries has been used.

The XML parser has been implemented by means of the CodeSynthesis XSD.

Some effort has been put into enhancing the performance of the software (e.g. vectorization, parallelization, etc.) reaching a reasonable execution speed (which of course does not depend only on the code). I am perfectly aware of the fact that the performance of the software can be improved with a better design and using more intrinsic features of C++. Nevertheless one of the main requirements was reusability and readibility of the code's structure has been kept as simple as possible knowing that C++ is one of the most difficult programming languages to read.

Validation

The orbit propagator has been validated by comparing its results with the ones generated by the evaluation version of FreeFlyer by a.i. solutions (evaluation version) and library Orekit 8.0. Some comparison results are distributed with the software package in the documentation. The plots released represent the difference between the generated orbit ephemerides in terms of orbital elements normalized to the semi-major axis and position-velocity state in orbital frame. The comparison has been done by considering first only the gravitational field and then the gravitational field plus single perturbation forces.

The attitude propagator has not been validated against any other tool.

The propagation results (dynamics, sensors' readings and actuator actions) have been throughly reviewed in a number of different simulations.

The events computation results have been validated by comparing the results with the ones generated by the evaluation version of FreeFlyer by a.i. solutions (evaluation version) and STK by AGI (free version).

Author — Over 19 years' experience in space system engineering with a specialization in space flight dynamics. This includes development and testing of guidance, navigation and control software, space operations and research in orbit control. This project stems from my great passion for astrodynamics and C++ software development.

Users — SpOCK is a domain specific software released with a package which requires the compilation of a C++ code and has no graphical user interface. The use of this tool requires some basic skills in compiling a code with GNU make and at least a basic knowledge of astrodynamics.

License — SpOCK is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation version 3.

Disclaimer — This software is the product of a free-time-project carried on by a single person and is provided 'as is' without warranty of any kind. There can be no warranty that:

(i) The software will meet your requirements

(ii) The software will be uninterrupted, timely, secure or error-free

(iii) The results that may be obtained from the use of the software will be effective, accurate or reliable

(iv) The quality of the software will meet your expectations

(v) Any errors in the software will be corrected

The author will try to maintain it and fix any found or reported bug in a resonable time but he cannot guarantee it.

Software and its documentation:

(i) Could include technical or other mistakes, inaccuracies or typographical errors

(ii) May be out of date, and the author makes no commitment to update such materials

2 System Requirements and Installation

Operating system — Linux Ubuntu 19 or higher

Build — GNU make (sudo apt-get install build-essential)

Compiler — gcc >= 7 supporting -std=c++17 (sudo apt-get install build-essential)

Libraries — Libraries usually not included by default in Linux Ubuntu distribution and to be installed are: libboost-all-dev, libxerces-c-dev, xsdcxx, gfortran, freeglut3, freeglut3-dev, mesa-utils, libsdl2*, libsoil*, doxygen, graphviz

These libraries (1125 MB of addtional disk space) will be installed automatically during the installation process.

Installation Steps — In root directory:

	make install_libs (first compilation or after OS upgrade)
	make install_atmo (first compilation or after OS upgrade)
	make install_mag (first compilation or after OS upgrade)
	If it is desired to use the Euler angles (not recommended) instead of the quaternions implementation of the attitude dynamics, the flag -DUSE_QUATERNION has to be deleted in the CCXX_FLAGS of the Make file
Build orbit propagator	make orbit
Build attitude propagator	make attitude
Build events computation	make events
	All the executable will be generated in the 'bin' folder. Command 'make clean' will delete all executables in 'bin' folder and folder 'obj'.
	Download file de438.bsp from SPICE library kernels (spk/planets) ftp (Sec. 8) and put it into folder data/cspice (file de438.bsp is too big for the software repository)
Remarks	
	Installation and run were tested successfully on different real and virtual machines (VMware on Windows 10) running Linux Ubuntu 19.04 or higher. It is possible but not guaranteed that the installation on Ubuntu < 19 or other Linux distributions could work.
	Coomands make install_atmo and install_mag are used to compile external Fortran libraries and could give some Fortran compilation warnings.

3 Files Structure

	The directories whose internal structure cannot be changed without changing also the C++ code and Make file are noted with a (!) sign.
Root directory	Main files for orbit and attitude propagation, and events computation.
bin	Executables for orbit and attitude propagation, and events computation
data (!)	All auxiliary files for orbit environment models and coordinates transformations
doc	All documentation
extlib (!)	All external libraries
input	All input files
lib (!)	All original code written by the author
objs	All compilation objects (.o files)
	This folder is generated when the software is compiled for the first time and can de deleted by typing on the terminal in the root directory 'make clean'
output	All output files

3.1 Data

Folder 'data' contains all the coefficients files required by the orbit environment models implemented and by the SPUCE library for the coordinate frames transformations. This folder can be put in any location by the user assumed that its internal structure is maintained intact. The absolute path of the folder is given by the user in the configuration parameters.

3.2 Libraries

Folder 'lib' contains all SpOCK's original source code. The internal structure of this folder as well as its relative position with respect to the make file has to be maintained intact. The C++ files are grouped in subfolder depending on their specific function (environment, spacecraft subsystems, etc.)

3.3 External Libraries

Folder 'extlib' contains all the external libraries used by the tool. The internal structure of this folder as well as its relative position with respect to the make file has to be maintained intact.

4 Configuration of Orbit and Attitude Propagation

All the parameters for orbit and attitude propagation has to be set up in the XML file /path_to_input/simparam_sample.xml released with the software package. The entries in this XML files is subject to the rules defined in the XML schema file simparam_schema.xsd. If an entry in the XML file is not compliant with the schema rules, when the software is launched an error message will appear telling where in the XML file is the false entry.

In the next sections all parameters and corresponding XML tags are described in detail. In file simparam_sample.xml comments have been added for each tag to avoid that the user has to refer continuously to this user guide. Following are some remarks for the use of the propagation parameters.

1. Spacecraft structure-parameters are required only if the panels model (Sec. 4.3.4) is used for the atmospheric drag and solar radiation perturbatio forces

2. An orbit ephemeris in the format described in Sec. 6 is required to run an attitude propagation. The propagation step has to be put equal to the step of the input ephemeris.

3. An attitude ephemeris in the format described in Sec. 6 is required to run an orbit propagation with the options to have the attitude given by an input ephemeris (Sec. 4.3.4). The propagation step has to be put equal to the step of the input ephemeris.

4. Maneuvers are the only elements which can be just added to the XML simulation parameters file without limits. To add a new subsystem (e.g. sensor, actuator) the software has to be modified (Sec. 9)

4.1 Spacecraft Structure and Properties

4.1.1 Structure

The spacecraft structure is defined relative to its impact on aerodynamic drag and solar radiation pressure. For this reason it consists of 6 faces whose position with respect to the spaceraft body-fixed coordinatesis defined by the position of the faces' aerodynamic and pressure center. This setup gives a large flexibility in doing accurate simulations with an economy of means.

Main XML tag	<SC_Faces>
6 faces	<Face name="Face_name">
Faces surface and material	<Area unit="m2">, <Material>
	Options for materials are Solar Array, MLI and Alluminium
Center of pressure	<cP_position unit="m">
	Position vector in spacecraft body-fixed coordinates system
Aerodynamic center	<cA_position unit="m">
	Position vector in spacecraft body-fixed coordinates system

4.1.2 Properties

Main XML tag	<SC_properties>
Mass and center of mass	<CoG>, <SC_mass unit="kg">, <CoG_pos unit="m">
	Position vector in spacecraft body-fixed coordinates system
Inertia matrix	<InertiaMatrix unit="kg*m2">
	Moments of inertia referenced to spacecraft body-fixed coordinates system
Coefficients	<Coefficients>
	Aerodynamic drag and solar radiation pressure coefficients. In case 'Panels' drag or SRP model type is used (Sec. 4.3.4), it is recommended to use value 1.0
Reference areas	<Areas>
	For aerodynamic drag and solar radiation pressure in case the 'RefArea' drag or SRP model type is used (Sec. 4.3.4)
	Only for orbit propagation
Spacecraft's dipole vector	<SC_dipole name="dipole_name" unit="Am^2">
	Dipole vector in spacecraft body-fixed coordinates system

4.2 Spacecraft Subsystems

The outputs of all spacecraft subsystems are explained in Sec. **??**. Spacecraft subsystems have the following generic definition in the XML schema

Main XML tag	<SensorsActuators name="subsystem_name">
Subsystem on/off	<subsystem_on>
	True or false

Constant parameters	<constparam name="param_name" unit="SI unit">
	Subsystem specification (e.g. thrust of propulsion unit)
	Fields 'name' and 'unit' are required
Auxiliary parameters	<auxparam name="auxparam_name" unit="SI unit">
	Parameter to refine subsystem specification (e.g. solar cells efficiency for solar panel)
	Fields 'name' and 'unit' are required
Operational limit	<opslimit name="opslimit_name" unit="SI unit">
	Limits of operations of subsystem (e.g. field of view of a camera)
Accuracy	<accuracy name="mean" unit="SI unit">, <accuracy name="std" unit="SI unit">
	Accuracy of a sensor or actuators in terms of mean and standard (e.g. magnetometer estimation accuracy)
	Sensors' readings and actuators' actions errors are generated by the C++ standard function 'normal_distribution'
	For a new sensor or actuator eventually to be implemented up to 3 couple (mean and standard deviation) of accuracy parameters can be defined
Subsystem integration	<SC2SYS_matrix>
	Rotation matrix from spacecraft's body-fixed to subsystem's coordinate system

4.2.1 Attitude Determination and Control

Sun camera	<SensorsActuators name="Sun Camera">, field 'name' cannot be changed
	<subsystem_on>
	<opslimit name="Camera FOV" unit="rad"> Entire field of view of the camera
	<accuracy name="mean" unit="rad">, <accuracy name="std" unit="rad">
	<SC2SYS_matrix>, in the simulation software the sensor Z-axis is the considered the main sensor axis i.e. the one representing the axis of the conical field of view
Earth camera	<SensorsActuators name="Earth Camera">, field 'name' cannot be changed
	The features implemented for the Earth camera are identical to the one of the Sun camera
6 coarse sun sensors	<SensorsActuators name="CSS_name">, field 'name' has to contain the word 'CSS'
	<subsystem_on>
	<constparam name="Nominal current at zero Sun incidence" unit="mA">
	<constparam name="Sensor signal noise" unit="mA"
	<opslimit name="CSS FOV" unit="rad"> Entire field of view of the sensor

	<accuracy name="mean" unit="rad">, <accuracy name="std" unit="rad">
	<SC2SYS_matrix>, in the simulation software the sensor Z-axis is the considered the main sensor axis i.e. the one representing the axis of the conical field of view
2 magnetometers	Only one magnetometer can be used in a simulation
	<SensorsActuators name="Magnetometer">, field 'name' cannot be changed
	<subsystem_on>
	<accuracy name="mean" unit="Nanotesla">, <accuracy name="std" unit="Nanotesla">
	<SC2SYS_matrix>
Rate sensors	<SensorsActuators name="Rate Sensor">, field 'name' cannot be changed
	<subsystem_on>
	<accuracy name="mean" unit="rad">, <accuracy name="std" unit="rad">
	<SC2SYS_matrix>
3 magneto torquers	The magnetotorquers are modelled as a block of 3 mutual perpendicolar magnetotorquers aligned to the axes of a right-handed coordinates system
	<SensorsActuators name="Magnetorquer">, field 'name' cannot be changed
	<subsystem_on>
	<constparam name="Conversion factor" unit="A*m^2/ms">, conversion factor from ontime to dipole produced by a magnetic torquer (m = ontime2dipole*ontime) [A*m^2/ms]
	<accuracy name="mean" unit="A*m^2">, <accuracy name="std" unit="A*m^2">
	<SC2SYS_matrix>
3 reaction wheels	<SensorsActuators name="Wheel_name">, field 'name' has to contain the word 'Wheel' The inertia of the wheels is included in the attitude dynamics
	<subsystem_on>
	<constparam name="Reaction wheel" unit="kg*m^2">, moment of inertia of the wheel [kg*m^2]
	<auxparam name="Init speed" unit="RPM">
	<accuracy name="mean speed" unit="RPM">, <accuracy name="std speed" unit="RPM">, accuracy (mean and standard deviation) of wheels speed measurements
	<accuracy name="mean torque" unit="Nm">, <accuracy name="std torque" unit="Nm">, accuracy (mean and standard deviation) of torque generated by wheels
	<SC2SYS_matrix>, in the simulation software the wheel Z-axis is considered the rotation axis

4.2.2 Orbit Control

2 propulsion systems <SensorsActuators name="Prop1>, <SensorsActuators name="Prop2>, field 'name' cannot be changed

<subsystem_on>

<constparam name="Thrust" unit="N">

<constparam name="Thrust resolution" unit="N">

<opslimit name="Maximum available thrust" unit="Ns">

<accuracy name="Propulsor dv accuracy mean" unit="%">, <accuracy name="Propulsor dv accuracy std" unit="%">, performance error

$\mid dv_{REAL} - dv_{CMD} \mid / dv_{CMD}$

<accuracy name="Attitude control accuracy mean" unit="rad">, <accuracy name="Attitude control accuracy std" unit="rad">

<SC2SYS_matrix>

4.2.3 Power

The power from the solar panels is computed based on the angle of incidence of the Sun on the panel and the efficiency.

3 solar panels <SensorsActuators name="panel_name">

<subsystem_on>

<constparam name="Surface" unit="m^2">

<constparam name="Voltage" unit="V">, used to compute the output current

<auxparam name="Solar cells efficiency" unit="-">

<opslimit name="FOV for Sun" unit="rad">

<SC2SYS_matrix>, the Z axis of the solar panel has to be perpendicular to its surface and pointing in the direction outside the spacecraft

4.3 Simulation Parameters

The simulation parameters are all those value required to run a simulation like initial states, inclusion of perturbation forces, etc.

Main XML tag <SimParameters>

4.3.1 Time parameters

The time format of the time parameters of the simulation is that of the XML schema duration type (e.g. P5Y2M10DT15H = 5 years + 2 months + 10 days + 15 hours, P60D = 60 days, PT10S = 10 s).

Main XML tag <durstep>

Integration step <simstep>

For attitude propagation this value has to be the same of the step of the loaded orbit ephemerides (e.g. 1 s, 10 s, etc.)

Integration duration <simduration>

For attitude propagation this value must have a value <= of the difference between the last and first time of the loaded ephemerides

4.3.2 Orbit Initial State

Main XML tag <ORB_initstate>

Initial time and state <Initime>, <Position unit="m">, <Velocity unit="m/s">

The initial time is UTC date and time in XML schema format (e.g. yyyy-mm-ddTHH:MM:SS)

The orbit state is given in ECI (Sec. B)

4.3.3 Attitude Initial State

Main XML tag <ATT_initstate>

Initial state <phi unit="deg">, <theta unit="deg">, <psi unit="deg">, <om_x unit="deg/s">, <om_y unit="deg/s">, <om_z unit="deg/s">

Epochs at each attitude propagation step is coincident with the epochs in the input orbit ephemeris

Angles phi, theta and psi represent an Euler rotation 321 with respect to the the ECI or RTN reference frame (Sec. B) depending on the value of <initstate_in_RTN> in <simoptions> (Sec. 4.3.4). If RTN is the chosen reference frame, <(phi,theta,psi) = (0,0,0)> means that $x_{SC} \equiv R$, $y_{SC} \equiv T$ and $z_{SC} \equiv N$. If ECI is the chosen reference frame, <(phi,theta,psi) = (0,0,0)> means that $x_{SC} \equiv x_{ECI}$, $y_{SC} \equiv y_{ECI}$ and $z_{SC} \equiv z_{ECI}$.

The angualar velocities om_x, om_y, om_z are given with respect to the spacecraft body-fixed reference system axes

4.3.4 Propagation Options

Main XML tag <simoptions>

Type of attitude initial state <initstate_in_RTN>

True or false

True: the attitude initial state given in tag <ATT_initstate> (Sec. 4.3.3) is in orbital RTN frame (Sec. B)

False: the attitude initial state will be considered given in inertial frame ECI (Sec. B)

Enable real time simulation <realtime>

True or false. Enable/disable real time option for hardware-in-the-loop (HIL) simulations

In case HIL is enabled, the output files will be written at each step of the propagation and not after the propagation loop is completed. The rationale is to have the possibility to check the results during a simulation which has a realtime step (tipically 1 s)

Waiting time for HIL <realtime_wait unit="s">

Waiting time for the execution of each integration step for hardware-in-the-loop simulations. Its value depends on the HIL simulation requirements and hardware.

Gravity gradient <ggrad_on>

	True or false. Enable/disable gravity gradient perturbation torque
	Only for attitude propagation
Magnetic perturbation	<mag_on>
	True or false. Enable/disable magnetic perturbation torque
	Spacecraft dipole (Sec. 4.1.2) interacting with Earth's magnetic field is used to compute the torque
	Only for attitude propagation
Solar radiation pressure	<srp_on>
	True or false. Enable/disable solar radiation pressure perturbation
	For orbit and attitude propagation
Atmospheric drag	<drag_on>
	True or false. Enable/disable atmospheric drag perturbation
	For orbit and attitude propagation
Gravity field order/degree	<nMAX>
	Positive integer number <= 120. Order and degree of gravity field model
	Only for orbit propagation
Third body	<sunmoon_on>
	True or false. Enable/disable third body perturbation
	Only for orbit propagation
	Sun and Moon third body perturbation are considered
Drag model type	<Drag_Model>
	Possible values are 'RefArea' or 'Panels'
	If RefArea is chosen the atmospheric drag perturbation is computed using reference area and drag coefficient given in Sec. 4.1.2
	If Panels is chosen, the perturbation is computed considering the properties of the spacecraft faces and the spacecraft attitude
	Option RefArea is only for orbit propagation, attitude propagation will use by default the Panels option
SRP model type	<SRP_Model>
	Possible values are 'RefArea' or 'Panels'
	If RefArea is chosen the solar radiation pressure perturbation is computed using reference area and drag coefficient given in Sec. 4.1.2
	If Panels is chosen, the perturbation is computed considering the properties of the spacecraft faces and the spacecraft attitude
	Option RefArea is only for orbit propagation, attitude propagation will use by default the Panels option
Attitude type	<AttitudeType>
	Attitude type during orbit propagation. Possible values are 'Fixed', 'RTN_fixed' and 'Ephemeris'

Fixed: the S/C keeps the attitude given in tag <ATT_initstate< (Sec. 4.3.3) fixed with respect to the inertial frame. This attitude can be given in inertial or RTN frame by using the <initstate_in_RTN> flag

RTN_fixed: the S/C keeps the attitude given in tag <ATT_initstate< (Sec. 4.3.3) fixed with respect to the RTN orbital frame. The attitude given in tag <ATT_initstate< is considered given in RTN frame (in this case tag <initstate_in_RTN> is not used)

Ephemeris: the attitude is read from a quaternion attitude ephemeris file (Sec. 6) with step equal to the chosen propagation step. The ephemeris file name is given in tag <Attitude_ephemeris< (Sec. **??**)

Only for orbit propagation

Enable attitude control <attctrl_on>

True or false

Only for attitude propagation

In the software only the if scope for the insertion of one or a number of different attitude controls is implemented. The insertion of an attitude control software is up to the user.

Attitude control type <AttCtrlType>

Possible values of this tag depends on the user's own implementation

Enable orbit control <orbctrl_on>

True or false

Only for orbit propagation

In the software only the if scope for the insertion of one or a number of different orbit controls is implemented. The insertion of an orbit control software is up to the user.

Orbit control type <OrbCtrlType>

Possible values of this tag depends on the user's own implementation

4.4 Input Files

This tag includes all the files required to run a simulation (input ephemerides, models coefficients, etc.).

Main XML tag <InputFiles>

Orbit ephemeris <Orbit_ephemeris>

Absolute path of orbit ephemeris file (Sec. 6) input for the attitude propagation (e.g. /path_to_ephem_folder/orbit_ephemeris.csv)

Only for attitude propagation

Attitude ephemeris <Attitude_ephemeris>

Absolute path of attitude ephemeris file (Sec. 6) optional input (Sec. 4.3.4) for the attitude propagation (e.g. /path_to_ephem_folder/attitude_ephemeris.csv)

Only for orbit propagation

Path of data folder <Data_path>

Absolute path of data folder (Sec. 3) (e.g. /path_to_data/data)

Planets ephemeris file <Planet_ephemeris>

Name of planet ephemeris .bsp file (e.g. de430.bsp). This file has to be placed into folder data/cspice

(e.g. the file's path will be /path_to_data/data/cspice/de430.bsp)

Earth orientation parameters <EOP_parameters>

Name of Earth orientation parameters .bpc file (e.g. earth_000101_200613_200322.bpc). This file has to be placed into folder data/cspice (e.g. the file's path will be /path_to_data/data/cspice/earth_000101_200613_200322.bpc)

PcK file <PCK_data>

Name of P_constants (PcK) SPICE kernel .tpc file (e.g. pck00010.tpc). This file has to be placed into folder data/cspice (e.g. the file's path will be /path_to_data/data/cspice/pck00010.tpc)

Leap second <Leap_second>

Name of leap seconds .tls file (e.g. naif0012.tls). This file has to be placed into folder data/cspice

(e.g. the file's path will be /path_to_data/data/cspice/naif0012.tls)

Gravity model <Gravity_model>

Options are 'GGM02C' and 'EIGEN-6S'

Each option corresponds to a gravity model coefficients file located in folder data/gravityfield (e.g. /path_to_data/data/gravityfield/GGM02C.gfc)

Atmosphere density model <Atmospheric_model>

Options are 'JB2008' and 'NRLMSISE-00'

JB2008: for using this model files DSTFILE.TXT, DTCFILE.TXT and SOLFSMY.TXT (Sec. 8) have to be present in folder data/atmosphere/JB2008

NRLMSISE-00: for using this model file CssiSpaceWeather_indices.txt (Sec. 8) has to be present in folder data/spaceweather

Magnetic field model <Magnetic_model>

Options are 'IGRF13' and 'WMM2020'

IGRF13: Coefficients are hardcoded in model's software

WMM2020: for using this model file WMM2020.COF (Sec. 8) has to be present in folder data/magneticfield/WMM

4.5 Output Files

This tag includes all products of a simulation (ephemerides, sensors readings, etc.)

Main XML tag <OutputFiles>

Orbit ephemeris <Orbit_ephemeris>

Absolute path of orbit ephemeris file (Sec. 6) output of the orbit propagation

Only for orbit propagation

Attitude ephemeris <Attitude_ephemeris>

Absolute path of attitude ephemeris file (Sec. 6) output of the attitude propagation. Two files will be generated in this same path, one with the name fixed with this tag (e.g. Attitude.csv) and with the attitude parameterized with Euler angles, and another file with the word '_Quaternions' added to this name (e.g. Attitude_Quaternions.csv) with the attitude parameterized with quaternions

Only for attitude propagation

Sensors' readings <Attitude_ephemeris>

Absolute path of attitude sensors' readings file (Sec. 6) output of the attitude propagation

Only for attitude propagation

Torques <Torques>

Absolute path of environmental and actuators torques file (Sec. 6) output of the attitude propagation. Two files will be generated in this same path, one with the name fixed with this tag (e.g. Torques.csv) and another file with the word '_EnvironmentRTN' added to this name (e.g. Torques_EnvironmentRTN.csv)

Only for attitude propagation

Accelerations <Accelerations>

Absolute path of environmental and actuators accelerations file (Sec. 6) output of the orbit propagation

Only for orbit propagation

4.6 Maneuvers

Attitude and orbit maneuvers can be added without any limitations in the simulation parameters XML file. Maneuvers have the following generic definition in the XML schema

Main XML tag <Maneuvers>

Maneuver type <Man name="maneuver_type">

Maneuver on/off <subsystem_on>

True or false. Enable/disable maneuver

Start time <init_time>

Positive real number representing the number of seconds from simulation start at which the maneuver has to be executed

Maneuver vector <ManVec unit="SI unit">

Maneuver vector depends on type of maneuver

4.6.1 Attitude Maneuvers

The available type of attitude maneuvers correspond to the implemented attitude actuators

Reaction wheels <Man name="WheelsManeuver">

Field 'name' cannot be changed

Maneuver vector components' units is rounds per minute (RPM)

Magnetotorquer <Man name="MagnetoManeuver">

Field 'name' cannot be changed

Maneuver vector components' are ontimes with units ms

Ontimes represent the time duration during which the actuation along a certain component of the spacecraft body-fixed frame is switched-on

Thrusters <Man name="ThrustersManeuver">

Field 'name' cannot be changed

Maneuver vector components' are ontimes with units ms

NOTE: this feature has not been implemented yet

4.6.2 Orbital Maneuvers

Orbital maneuvers are classified according to the type of thrust (impulsive or continuous), the coordinate system of the maneuver vector (RTN or spacecraft body-fixed) and the propulsion system by which they are executed (2 propulsion units are implemented). All possible combinations are possible. The type of maneuver and propulsion system are identified by the tag name.

Type of maneuver <Man name="prop_maneuver_type_name">

Maneuver vector components' are dv (velocity variation) with units m/s

Field 'name' cannot be changed. Possible names are

ImpulsiveManeuverProp1_RTN

ContinuousdManeuverProp1_Body

ContinuousManeuverProp1_RTN

ContinuousManeuverProp1_Body

ImpulsiveManeuverProp2_RTN

ContinuousdManeuverProp2_Body

ContinuousManeuverProp2_RTN

ContinuousManeuverProp2_Body

Properties of Prop1 and Prop2 propulsion units are defined in Sec. 4.2.2

ImpulsiveManeuverProp#_RTN is an impulsive maneuver vector given in the orbital frame RTN frame centered in the spacecraft centre of mass and executed by Prop3 (# is 1 or 2). The propulsion system characteristics given in Sec. 4.2.2 are NOT taken into account for the inclusion of the maneuver in the propagation.

ImpulsiveManeuverProp#_Body is an impulsive maneuver vector given in the spacecraft body-fixed frame. The attitude of the spacecraft (Sec. 4.3.4) is taken into account for the inclusion of the maneuver in the propagation. The propulsion system characteristics given in Sec. 4.2.2 are NOT taken into accout for the inclusion of the maneuver in the propagation because the dv generation is considered istantenuous (and then not depending on the impulsive thrust capability of the propulsion unit).

ContinuousManeuverProp#_RTN is a continuous maneuver vector given in RTN frame. The propulsion system characteristics given in Sec. 4.2.2

are taken into account for the inclusion of the maneuver into the propagation.

ContinuousManeuverProp#_Body is a continuous maneuver vector given in the spacecraft body-fixed frame. Both propulsion system characteristics given in Sec. 4.2.2 and attitude of the spacecraft (Sec. 4.3.4) are taken into account.

Each time a maneuver is executed during a simulation, the value of the commanded dv is subtracted from the total available dv budget (see Sec. 4.2.2).

Different type of maneuvers (impulsive or continuous) are allowed for the same propulsion system in the same simulation. Impulsive and continuous maneuvers can be superimposed but two continuous maneuvers cannot. By giving the desired dv for a continuous maneuver and the propulsion system characteristics, the software computes the maneuvers duration. If the start time of a continuous maneuver results before final time of a previous continuous maneuver amessage error during the run of the simulation will appear and the execution will be interrupted.

5 Configuration of Events Computation

All the parameters for the events computation has to be set up in the XML file /path_to_input/eventsparam_sample.xml released with the software package. The entries in this XML files is subject to the rules defined in the XML schema file simparam_schema.xsd. If an entry in the XML file is not compliant with the schema rules, when the software is launched an error message will appear telling where in the XML file is the false entry.

In the next sections all parameters and corresponding XML tags are described in detail. In file eventsparam_sample.xml comments have been added for each tag to avoid that the user has to refer continuously to this user guide. Following are some remarks for the use of the propagation parameters.

An orbit ephemeris in the format described in Sec. 6 is required to run a computation.

5.1 Computation Parameters

Main XML tag <CompParameters>

5.1.1 Time parameters

The time format of the time parameters of the simulation is that of the XML schema duration type (e.g. P5Y2M10DT15H = 5 years + 2 months + 10 days + 15 hours, P60D = 60 days, PT10S = 10 s).

Main XML tag <durstep>

Integration step <simstep>

This value has to be the same of the step of the loaded orbit ephemerides (e.g. 1 s, 10 s, etc.)

Integration duration <simduration>

This value must have a value <= of the difference between the last and first time of the loaded ephemerides

5.1.2 Payload

An elliptical field of view (FOV) of the payload can be defined. The angles given represent the half-angles of the FOV (e.g. if cross and along angles have a value of 30 deg, the FOV will be a cone with circular footprint and aperture equal to 60 deg)

Main XML tag <Payload>

Field of view <FOV_cross unit="deg">, <FOV_along unit="deg">

The cross section defines the half-angle of the FOV perpendiculat to the spacecraft ground-track

The along section defines the half-angle of the FOV along the spacecraft ground-track

5.1.3 Number of Spacecraft

The computation of events has been devised for a satellites constellation. The orbit ephemerides considered for a costellation has always to have the prefix S#-P#_ in their name. To consider only one spacecraft and one ephemeris with the name given by <Orbit_ephemeris_rootname> in the input files (Sec. 5.4), all values of SC_* and PL_* has to be 1.

Main XML tag <Spacecraft>

Spacecraft number <SC_start>, <SC_end>

Integer value >= 1

Orbit plane number <PL_start>, <PL_end>

Integer value >= 1

5.1.4 Computation Options

Main XML tag <Compoptions>

Target contacts <TGs_on>

True or false. Enable/disable computation of contacts between payload footprint and targets (TG) location

Ground station contacts <GSs_on>

True or false. Enable/disable computation of contacts with ground station (GS)

Targets grid <TGs_grid_on>

True or false. Enable/disable computation of payload contacts with targets grid

Eclipse <Eclipse_on>

True or false. Enable/disable computation of umbra/penumbra start and end times

5.2 Targets

Main XML tag <TGs>

5.2.1 Targets Grid

Main XML tag <TGs_grid>

Grid definition <minlon unit="deg">, <maxlon unit="deg">, <minlat unit="deg">, <maxlat unit="deg">, <gridstep unit="deg">

Longitude is defined in the interval [-180, 180] and latitude in the interval [-90,90]

The grid is delimited by the great circles passing for the couples of geolocations (minlon,minlat) - (maxlon,minlat), (minlon,minlat) - (minlon,maxlat), (minlon,maxlat) - (maxlon,maxlat), (maxlon,minlat) - (maxlon,maxlat). The geolocation of each target on the grid is given by (lon, lat) = (minlon + k*gridstep, minlat + h*gridstep) with k, h $\in \mathbb{N}^+$ and minlon + k*gridstep <= maxlon, minlat + h*gridstep <= maxlat

in case maxlon ¡ minlon or maxlat ¡ minlat or gridstep ¿ maxlon - minlon or gridstep ¿ maxlat - minlat the program will exit with an error message.

5.2.2 Targets List

Main XML tag <TGs_list>

List definition <TG name="Target_name">, <lon unit="deg">, <lat unit="deg">, <alt unit="m">

Longitude is defined in the interval [-180, 180] and latitude in the interval [-90,90]

Target altitude is defined by a real number >= 0. NOTE: this feature has not been implemented yet

5.3 Ground Stations

Main XML tag <GSs>

List definition <GS name="GS_name">, <lon unit="deg">, <lat unit="deg">, <alt unit="m">, <minelev unit="deg">

Longitude is defined in the interval [-180, 180] and latitude in the interval [-90,90]

GS altitude is defined by a real number >= 0. NOTE: this feature has not been implemented yet

The minimum elevation determines the elevation to compute start and end time of the ground station pass

5.4 Input Files

This tag includes all the files required for the computation of the events.

Main XML tag <EventsInputFiles>

Orbit ephemerides folder <Orbit_ephemeris_path>

Absolute path of folder containing orbit ephemerides files (Sec. 6) input for the event computations (e.g. /home/dir1/dir2/ephem_folder)

Ephemerides files names <Orbit_ephemeris_rootname>

If a single spacecraft is considered (all values in <Spacecraft> are 1 Sec. 5.1.3), this is the complete name of the ephemeris file to be considered (e.g. if <Orbit_ephemeris_path> is path_to_ephem_folder and <Orbit_ephemeris_rootname> is ephemeris.csv, the only file considered for the computation of events will be /path_to_ephem_folder/ephemeris.csv)

If a constellation is considered (any value in <Spacecraft> is > 1 Sec. 5.1.3), this is the orbit ephemeris name without the prefix S#-P#_ (e.g. if in folder /path_to_ephem_folder there are ephemerides S1-P1_ephemeris.csv, S2-P1_ephemeris.csv, S1-P2_ephemeris.csv, S2-P2_ephemeris.csv and all thse ephemerides have to be used, the values of SC_* and PL_* in tag <Spacecraft> will have value 2 and <Orbit_ephemeris_rootname> will be ephemeris.csv)

Path of data folder <Data_path>

Absolute path of data folder (Sec. 3) (e.g. /path_to_data/data)

Planets ephemeris file <Planet_ephemeris>

Name of planet ephemeris .bsp file (e.g. de430.bsp). This file has to be placed into folder data/cspice

(e.g. the file's path will be /path_to_data/data/cspice/de430.bsp)

Earth orientation parameters <EOP_parameters>

Name of Earth orientation parameters .bpc file (e.g. earth_000101_200613_200322.bpc). This file has to be placed into folder data/cspice (e.g. the file's path will be /path_to_data/data/cspice/earth_000101_200613_200322.bpc)

PcK file <PCK_data>

Name of P_constants (PcK) SPICE kernel .tpc file (e.g. pck00010.tpc). This file has to be placed into folder data/cspice (e.g. the file's path will be /path_to_data/data/cspice/pck00010.tpc)

Leap second <Leap_second>

Name of leap seconds .tls file (e.g. naif0012.tls). This file has to be placed into folder data/cspice

(e.g. the file's path will be /path_to_data/data/cspice/naif0012.tls)

5.5 Output Files

This tag includes all products of the events computation

Main XML tag <EventsOutputFiles>

Targets contacts <TG_contacts>

Absolute path of file (Sec. 6) output of the computation of the contacts between payload footprint and targets. Two files are generated in this same path, one with the name fixed with this tag with all the contacts for all the satellites (in case of more than one satellites) and one for each satellite with names S#-P#_TG_contacts

File generated only if the targets contacts computation is enabled

Ground station contacts <GS_contacts>

Absolute path of file (Sec. 6) output of the computation of ground station contacts. Two files are generated in this same path, one with the name fixed with this tag with all the contacts for all the satellites (in case

of more than one satellites) and one for each satellite with names S#-P#_GS_contacts

File generated only if the ground station contacts computation is enabled

Eclipse times <Eclipse_times>

Absolute path of file (Sec. 6) output of the computation of umbra/penumbra times. Two files are generated in this same path, one with the name fixed with this tag with all the contacts for all the satellites (in case of more than one satellites) and one for each satellite with names S#-P#_GS_contacts

File generated only if umbra/penumbra times computation is enabled

6 Inputs, Outputs and Formats

All input ephemerides and output files are comma-separated values (CSV) files.

6.1 Sensors Output

Sun and Earth cameras Outputs of Sun and Earth cameras are Azimuth and Elevation angles in the camera frame in unit [10^{-2} [deg] and two flags for capture and detection of Sun and Earth. The values of these flags are at the current development stage of the software constant and hardcoded. A logic for determining their value can be implemented for HIL simulations.

Coarse sun sensor Output of the coarse Sun sensor is the current [A] produced by the sun arrays composing the sensor

Magnetometer Output of the magnetometer is the estimated Earth magnetic field vector [Nanotesla] in the magnetometer frame

Rate sensor Output of the rate sensor is the spacecraft estimated angular rotation vector [10^{-2} [deg] in the sensor frame

6.2 Actuators Output

Magnetotorquer Output of the magnetotorquer is the torque vector [Nm] produced by the actuator in the spacecraft body-fixed frame

Reaction wheel Output of the reaction wheel is the torque vector [Nm] produced by the actuator in the spacecraft body-fixed frame

6.3 Solar Panels Output

Outputs of the solar panel are the power [W] and current [A] generated by the solar panel

6.4 Orbit Propagation

6.4.1 Input

Attitude ephemerides If the option to have an attitude ephemeris as input is selected (Sec. 4.3.4), the format has to be the one described in Sec. 6.5 using quaternions

6.4.2 Output

Orbit ephemerides

The orbit propagation generates an orbit ephemeris file without header and a second file containing perturbation accelerations and eventual orbit maneuvers dvs with header

Step, GPS Time [s], X_{ECI}, Y_{ECI}, Z_{ECI}, VX_{ECI}, VY_{ECI}, VZ_{ECI}, X_{ECEF}, Y_{ECEF}, Z_{ECEF}, VX_{ECEF}, VY_{ECEF}, VZ_{ECEF}

where X, Y, Z are position vector components and VX, VY, VZ velocities

Accelerations and dvs

Accelerations and dv vectors components are given in the orbital RTN frame

GPS Time [s], Grav_R, Grav_T, Grav_N, SunMoon_R, SunMoon_T, SunMoon_N, SRP_R, SRP_T, SRP_N, Drag_R, Drag_T, Drag_N, Acc_R, Acc_T, Acc_N, dv_R, dv_T, dv_N

where Grav_*, SunMoon_*, SRP_*, Drag_*, Acc_* and dv_* are respectively the RTN components of the gravity, third body, solar radiation pressure, atmospheric drag and total acceleration vectors, and of the dv vector

6.5 Attitude Propagation

6.5.1 Input

The attitude propagation requires as input an orbit ephemeris in the format described in Sec. 6.4

6.5.2 Output

The attitude propagation generates two attitude ephemerides files (angles and quaternions), a sensor readings file and a torques file (environment and actuators).

Attitude ephemerides - Angles

GPS Time [s], ϕ, θ, ψ, ω_x, ω_y, ω_z

where angles phi, theta and psi represent an Euler rotation 321 with respect to the ECI or RTN reference system depending on the value of <initstate_in_RTN> in <simoptions> (Sec. 4.3.4) and the angualar velocities ω_x, ω_y, ω_z are given with respect to the spacecraft body-fixed reference system axes

Attitude ephemerides - Quaternions

GPS Time [s], q_1, q_2, q_3, q_4, ω_x, ω_y, ω_z

where q_* are the attitude quaternion components and the angualar velocities ω_x, ω_y, ω_z are given with respect to the spacecraft body-fixed reference system axes

Sensor readings

GPS Time, CSS_1, CSS_2, CSS_3, CSS_4, CSS_5, CSS_6, SunCam Az, SunCam El, SunCapture, SunDetection, EarthCam Az, EarthCam El, EarthCapture, EarthDetection, $\text{MagX}_\text{magframe}$, $\text{MagY}_\text{magframe}$, $\text{MagZ}_\text{magframe}$, MagR_RTN, MagT_RTN, MagN_RTN, $\text{RateX}_\text{sensframe}$, $\text{RateY}_\text{sensframe}$, $\text{RateZ}_\text{sensframe}$, WheelSpeedX, WheelSpeedY, WheelSpeedZ, SolarPanels1 Pow, SolarPanels1 Curr, SolarPanels2 Pow, SolarPanels2 Curr, SolarPanels3 Pow, SolarPanels3 Curr, Eclipse

All quantities in the sensor readings file are in the format explained in Sec. 6.1

The flag 'Eclipse' is 0 when the spacecraft is in the Sun and 1 when the spacecraft is in the shadow

Torques GPS Time [s], TenvX, TenvY, TenvZ, TenvR, TenvT, TenvN, TmX, TmY, TmZ, TwX, TwY, TwZ, TactX, TactY, TactZ, Mag. ontimes X, Mag. ontimes Y, Mag. ontimes Z, WheelCTRL X, WheelCTRL Y, WheelCTRL Z

where TenvX, TenvY, TenvZ and TenvR, TenvT, TenvN are the total orbit environment perturbation torques respectively in spacecraft body-fixed and RTN frames, Tm*, Tw*, Tact* are the magnetotorquers, wheels and total actuators-generated torques in spacecraft body-fixed, Mag. ontimes and WheelCTRL are the magnetotorquers' ontimes and wheels speeds commanded by an eventual attitude controller (to be implemented by the user or coming from a controller in HIL simulations)

GPS Time [s], GravR, GravT, GravN, MagR, MagT, MagN, SRP_R, SRP_T, SRP_N, DragR, DragT, DragN, T_R, T_T, T_N

where Grav*, Mag*, SRP*, Drag* and T* are respectively the gravity gradient, magnetic, solar radation, atmospheric drag and total perturbation torques vectors components in RTN frame

6.6 Events Computation

6.6.1 Input

The events computation requires as input an orbit ephemeris in the format described in Sec. 6.4

6.6.2 Output

The events computation generates only the requested output files (payload-target contacts, ground station contacts, umbra/penumbra times)

Targets contacts The file containing all the contacts has the format

TG, Epoch UTC, GPS time [s], Duration [s], El. in, El. out, Epoch UTC Max. El., Max. El. [deg], Az. in [deg], Az. out [deg], Az. Max El. [deg], lon [deg], lat [deg], Orbital plane, Spacecraft

where TG is the target name, Epoch UTC and GPS time [s] is the start epoch of the contacts, Duration is the contact time duration in seconds, Az. in, El. in, Az. out, El. out are the azimuth and elevation of the spacecraft at the start and end of the contact, Epoch UTC Max. El. and Az. Max El. are the epoch and azimuth when the spacecraft has maximum elevation (Max El.) over the target, lon, lat are the geodetic coordinates of the target and Orbital plane, Spacecraft are the orbital plane and spacecraft numbers.

The files containing all contacts for each spacecraft are identical to the one for all spacecraft except for the Orbital plane and Spacecraft numbers that are missing in the file but are in the file's name

GS contacts The file containing all the contacts has the format

GS, AOS UTC, LOS UTC, AOS [GPS secs], Duration [m], Epoch UTC Max. El., Max Elevation [deg], Az. AOS [deg], Az. LOS [deg], Az. Max El. [deg], lon [deg], lat [deg], Orbital plane, Spacecraft

where AOS and LOS are the start and end of the contact, the duration of the contacts is in minutes and all the other parameters have the meaning already explained for the targets contacts

The files containing all contacts for each spacecraft are identical to the one for all spacecraft except for the Orbital plane and Spacecraft numbers that are missing in the file but are in the file's name

Eclipse times — Penumbra start UTC, Umbra start UTC, Umbra end UTC, Penumbra end UTC, Umbra duration [m], Penumbra duration [m], Orbital plane, Spacecraft

All epochs are UTC and the durations are in minutes

The files containing all umbra/penumbra times for each spacecraft are identical to the one for all spacecraft except for the Orbital plane and Spacecraft numbers that are missing in the file but are in the file's name

7 Run

Assuming that the executables are in the 'bin' folder and the propagation and events computations parameters files (Sections 4 and 5) are in folder 'input', run the executables in linux terminal typing the following commands

Orbit propagation — ./path_to_bin/bin/OrbitPropagator /path_to_input/simparam_sample.xml

Attitude propagation — ./path_to_bin/bin/AttitudePropagator /path_to_input/simparam_sample.xml

Events computation — ./path_to_bin/bin/EventsPropagator /path_to_input/eventsparam_sample.xml

7.1 Run with Sample Configuration Files

In the release two sample configuration files simparam_sample.xml and eventsparam_sample.xml. In simparam_sample.xml the configuration for a small satellite (150 kg) mission in low Earth orbit (LEO) is given. In eventsparam_sample.xml an example of some ground stations and target list and grid is given. An example of input and output .csv files using these configuration files is also released in the 'input' and 'output' directories.

8 Maintenance

Eigen library — The latest version of the Eigen library in folder extlib/Eigen can be downloaded here `http://eigen.tuxfamily.org`

Version currently in use: 3.3.7 released 18.12.2018 and maintained on `https://bitbucket.org/eigen/eigen`

SPICE library kernels — The SPICE library kernels in folder data/cspice have to be maintained up-to-date and can be retrieved here `ftp://naif.jpl.nasa.gov/pub/naif/generic_kernels/`

XML parser — The latest version of the XML parser C++ library CodeSynthesis XSD can be downloaded here `https://www.codesynthesis.com/products/xsd/`

Version currently in use: 4.0.0 released 31.07.2014

xerces library — The latest version of C++ library xerces used by CodeSynthesis XSD can be downloaded here `http://xerces.apache.org/xerces-c/download.cgi`

To update replace folder Pextlib/libxsd/xsd/cxx/xml/xercesc with the one src/xercesc found in the downloaded packet.

Version currently in use: xerces-c-3.2.2

JB2008 model

Files in folder data/atmosphere/JB2008 have to be maintained up-to-date `http://sol.spacenvironment.net/~JB2008/indices.html`

In case there is an update of Fortran .for model files, or the attitude and orbit propagators are compiled for the first time or the operating system is upgraded, run in root directory 'make install_atmo'. This command will delete all .o files in folder extlib/Atmosphere/JB2008 and recompile the Fortran software with line commands gfortran -c JB2008.f -o JB2008.o and gfortran -c Readfiles.f -o Readfiles.o (see also Sec. 2)

NRLMSISE-00 model

The Fortran implementation of the model has been retrieved here `https://ccmc.gsfc.nasa.gov/pub/modelweb/atmospheric/msis/nrlmsise00/`

In case there is an update of Fortran .for model files, or the attitude and orbit propagators are compiled for the first time or the operating system is upgraded, run in root directory 'make install_atmo'. This command will delete all .o files in folder data/atmosphere/NRLMSISE-00/FORTRAN and recompile the library with line command gfortran -c -std=legacy nrlmsise00_sub.for -o nrlmsise00_sub.o

Space weather data file data/spaceweather/CssiSpaceWeather_format.txt can be downloaded here `https://celestrak.com/SpaceData/`. Notice that the name of file CssiSpaceWeather_format.txt must not be changed

IGRF model

The most recent version of the international geomagnetic reference field (IGRF) model can be retrieved here `https://www.ngdc.noaa.gov/IAGA/vmod/igrf.html`

In case there is an update of Fortran .f model files, or the attitude and orbit propagators are compiled for the first time or the operating system is upgraded, run in root directory 'make install_mag'. This command will delete all .o files in folder extlib/MagneticField/IGRF/FORTRAN delete file igrf13.o and recompile the library with line command gfortran -c igrf13.f -o igrf13.o

Version currently in use is IGRF13.

WMM model

The most recent version of the world magnetic model (WMM) can be retrieved here `https://ngdc.noaa.gov/geomag/WMM/DoDWMM.shtml`.

In case there is an update of Fortran .for model files, or the attitude and orbit propagators are compiled for the first time or the operating system is upgraded, run in root directory 'make install_mag'. This command will delete all .o files in folder extlib/MagneticField/WMM/FORTRAN delete file geomag.o and recompile the library with line command gfortran -c geomag.for -o geomag.o

Version currently in use is WMM2020.

9 Modifications and Customization

9.1 XML Parser

Any software modification which requires a new tag in the simulation parameters XML files (e.g. new propagation option, new subsystem, etc.) requires a modification of the CodeSynthesis XML parser by means of the following procedure

1. Modify the schema simparam_schema.xsd

2. Install xsdcxx (W3C XML Schema to C++ Compiler) with sudo apt-get install xsdcxx if it has not been already installed
3. In the linux terminal go into the folder containing the XML schema simparam_schema.xsd
4. Enter in terminal: xsdcxx cxx-parser –xml-parser xerces simparam_schema.xsd –¿ files simparam_schema-pskel.hxx and simparam_schema-pskel.cxx are generated
5. Enter in terminal: xsdcxx cxx-parser –generate-print-impl simparam_schema.xsd –¿ files simparam_schema-pimpl.hxx and simparam_schema-pimpl.cxx are generated
6. Enter in terminal: xsdcxx cxx-parser –generate-test-driver –root-element simparam simparam_schema.xsd –¿ file simparam_schema-driver.cxx is generated
7. Change names of simparam_schema-pskel.hxx and simparam_schema-pskel.cxx in simparam_schema-pskel.h and simparam_schema-pskel.cpp and copy and past them into folder lib/IO/XML_Parser overwriting the ones already there
8. In file simparam_schema-pskel.cpp change line #include "simparam_schema-pskel.hxx" into #include "simparam_schema-pskel.h"
9. Compare files simparam_schema-pimpl.hxx and simparam_schema-pimpl.cxx with files simparam_schema-pimpl.h and simparam_schema-pimpl.cpp in folder lib/IO/XML_Parser
10. Modify manually files simparam_schema-pimpl.h and simparam_schema-pimpl.cpp in folder lib/IO/XML_Parser accordingly (new files simparam_schema-pimpl.hxx and simparam_schema-pimpl.cxx give an example of implementation of the parser according to the modifications introduced in simparam_schema.xsd)
11. Compare file simparam_schema-driver.cxx with function XML_parser in file lib/IO/IO_utils.cpp and if required modify manually function XML_parser
12. Rebuild the software

9.2 Spacecraft Subsystems

If the implementation of one or more new subsystems is required take into account that the total number of subsystems (in the current implementation sensors, actuators and solar panels) is limited to 30. If a larger number of sensor/actuators is required the dimension of class 'simparam_pimpl' member array 'SC::SYS_params SensAct_prms_in' in file lib/IO/XML_Parser/simparam_schema-pimpl.h has to be changed.

The number of subsystem 'accuracy' entries is limited to 6 (3*mean + 3*std). To change this limit modifications have to be made in the XML schema simparam_schema.xsd as well as in file lib/Spacecraft/Subsystem.cpp

Capture and detection of Sun and Earth cameras (Sec. 6.1) are at the current development stage of the software constant and hardcoded. A logic for determining their value can be implemented for more detailed software simulation or HIL simulations.

The implementation of a new subsystem is straightforward by creating a new SUBSYS derived class (files lib/Spacecraft/Subsystem.*).

In the 'main' files of the orbit and attitude propagator it is indicated with a commented example where to put orbit and attitude control functions developed by the user.

9.3 HIL

As for the orbit and attitude control, also for hardware-in-the-loop simulations in the 'main' files of the orbit and attitude propagator an if scope has been inserted with an interface function to be developed by the user.

10 Documentation

Documentation about the orbit environment model used and about the SPICE library has been included.

Among the documentation are included Valgrind's profiling products which can be viewed with kcachegrind (e.g. in terminal type kcachegrind callgrind.out.OrbitPropagatorAllPerturbations_1day) and plots pertaining to the orbit propagator validation

To generate developer documentation with doxygen it is required to put in file doc/UserGuide/doxydoc/Doxyfile at the entry INPUT the correct path of the source code.

11 Next Developments

The author has the plan (but no schedule for the actuation of this plan) to implement new objects and features.

Epichs Satellite constellation and spacecraft formation objects.

3D visualization of spacecraft attitude during simulations.

Features Coupling of orbit and attitude propagation.

Implementation of 3 indipendent magnetotorquers and a 4^{th} wheel.

Star camera model.

Improvements Make the software more robust (e.g. divide by zero, inputs editing, etc.).

Appendix

A Acronyms

AOS	Acquisition of Signal
CSS	Coarse Sun Sensor
CSV	Comma-Separated Values
ECEF	Earth-Fixed-Earth-Centered
ECI	Earth-Centered Inertial
EOP	Earth Orientation Parameters
FOV	Field of View
GS	Ground Station
HIL	Hardware-Iin-the-Loop
IGRF	International Geomagnetic Reference Field
ITRF	International Terrestrial Reference Frame
ITRS	International Terrestrial Reference System
LEO	Low Earth Orbit
LOS	Loss of Signal
OS	Operating system
RTN	Radial-Tangential-Normal
RPM	Rounds Per Minute
S/C	Spacecraft
SI	International System of Units
SpOCK	Spacecraft Orbital Computations Kit
SRP	Solar Radiation Pressure
TG	Target
WMM	World Magnetic Model

B Reference frames

The reference frames used are

Spacecraft body-fixed The geometry and masses displacement of the spacecraft with respect to the spacecraft body-fixed coordinate frame is defined in the propagator parameters XML file (Sec. 4.1.2) by fixing the center of mass position vector and the inertia matrix.

In the current implementation of the software the origin of the spacecraft body-fixed coordinate frame is assumed to be coincident with the spacecraft's center of mass.

Spacecraft subsystems Each spacecraft subsystem-fixed coordinate frame is defined relative to the spacecraft body-fixed frame by fixing in the propagator parameters XML file (Sec. 4.2) the rotation matrix from the spacecraft to the subsystem frame.

ECI The Earth-centered inertial coordinate frame used is the J2000.

ECEF The Earth-fixed-Earth-centered (ECEF) used is the ITRF93 international terrestrial reference frame realization of the international terrestrial reference system.

Orbital frame The orbital frame RTN is centered in the spacecraft centre of mass with R pointing from the center of the Earth to the spacecraft center of mass, N in the direction of the orbit angular momentum and T = NxR to close the right-handed coordinates system.

Index