Paweł ZAGÓRSKI, Wojciech GREGA
AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY,
FACULTY OF ELECTRICAL ENGINEERING, AUTOMATICS, COMPUTER SCIENCE AND BIOMEDICAL ENGINEERING

Aerospace Blockset for Xcos – Open source tool for aerospace systems numerical computation and simulation

Abstract

Aerospace Blockset for Xcos is a free, open and extendible software tool for aerospace systems simulations. It is a part of the open source Scilab/Xcos simulation environment. It was envisioned as a tool that fills the gap between the free, single-purpose aerospace software (e.g.. Space Trajectory Analysis, etc.), and the very expensive professional tools (eg. AGI-STK, Matalb-Simulink). Aerospace Blockset provides wide capabilities and is available for free even for commercial use.

Keywords: orbit propagation, attitude estimation, simulations, space research.

1. Introduction

Scilab [1] is free and open source software for numerical computation providing a powerful computing environment for engineering and scientific applications. It is complemented with Xcos, which allows for diagrammatic programming and is intended for design of hybrid dynamical systems models. Those two tools are free alternative to commercial MATLAB/ SIMULINK package. They offer many functionalities which can be used for professional purposes (for example [2]). Due to the open source nature Scilab/Xcos also allows for adding third party toolboxes. Most of them are also open source and free to use.

One of the most interesting additional toolboxes for Scilab is CelestLab. It is an open source aerospace library containing hundreds of aerospace functions. It is designed, developed and used by the professional staff of the French Space Agency - CNES (Centre National d'Etudes Spatiales, The French Space Agency). The library was tested against the commercial software and real space mission data, therefore it is highly reliable. The only downside of this toolbox is that it does not include Xcos blocks. Therefore, to use CelestLab a substantial programming skills are needed. While solving real-life problem with CelestLab one will have to essentially write a computer program using Scilab script language. As this is a good solution for professionals it may be troublesome when used in education environment [3]. If some of CelestLab functionalities where available in form of Xcos blocks it would be possible to create scenarios using diagrammatic programming language. In such a diagram not only the end goal of performing desired calculations is achieved, but also it is done in more intuitive way. Looking at a diagram immediately allows to understand what operations are necessary to come up with the result, and what is the relationship between them. This would allow researchers and students to focus on the nature of the aerospace engineering problem, and to neglect details of implementation.

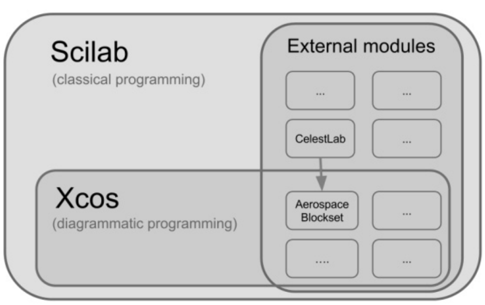

Fig. 1. Relationship between Scilab, Xcos and Aerospace Blockset

Seeing this potential the Authors of this paper took opportunity to develop the Aerospace Blockset (AB). The module is a set of 40 functional Xcos blocks (as for version 2.1) encapsulating many of the CelestLab functionalities into Xcos diagramatic programming tool, that is in turn part of Scilab environment. Figure 1 shows relationship between AB, Scilab/Xcos and CelestLab library. All of the developed functional blocks can be connected indicating data flow, and in that way solutions to complex aerospace problems can be found. Additionally, when user is required to input manually numerical parameters it is possible in many cases to do that using a dedicated dialogue box. Explanations of the parameters unit, meaning and allowable range further decreases the risk of error. Although the blockset does not yet allow to perform all the operations offered by the CelestLab library, it already makes possible to perform complicated simulations. Its diverse functionalities include but are not limited to: propagating orbits of celestial bodies and artificial satellites, conversions of reference and time frames, environmental models (Earth magnetic field, solar pressure, atmospheric drag, etc...), ground station visibility, unit conversions, attitude dynamics and quaternion algebra. But most notably, it allows building diagram that answer questions interesting also for non-professionals such as: "When will the International Station be visible from my home city?", "What is the current distance from the Earth to the Mars?", or "How large is the force that my satellite will be subjected to, due to the Solar Radiation pressure?"

2. Description

Scilab environment contains many other compatible blocksets (FEM simulations, statistical analysis, etc.) which further extends potential applications of AB. It is also possible modify it freely and write user-defined blocks to extend the capability and customise the tool for own needs. Users who develop new functionalities are free to either keep them for themselves or contact the developers and include them in future releases.

Its availability, capability, extendibility makes it a great tool for aerospace research, education (e.g. educational CubeSat projects [9]). Additionally, as the blockset is based on diagrammatic programming concept user can not only perform complex simulations, but also better imagine and understand the nature of simulated phenomenon by studying relationship between the functional blocks.

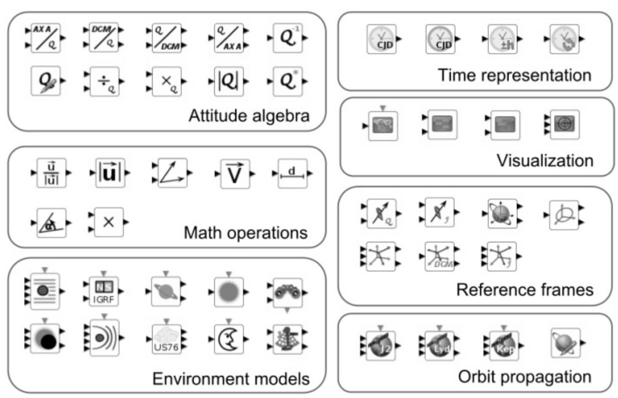

Fig. 2. AB blocks sorted by feature

Blockset features can be divided into seven functional groups, as shown in Figure 2. Attitude algebra blocks allow conversion between common attitude representations, namely quaternions, Direction Cosine Matrisces (DCM), and Euler axis and angle. They also provide algebraic quaternion operations that are not included in Xcos itself. Mathematical operation blocks enable converting between physical units, calculating vector products and measuring angles between vectors and distances in 3D space. Environmental models provide simulations of space conditions, such as position of the Sun, the Moon and planets [7], solar eclipses, direct line visibility, atmospheric pressure [5], atmospheric drag, solar radiation pressure and geomagnetic field [8]. Another groups of blocks enables converting between different time scales and spacial reference frames used in aerospace engineering. It is also possible to visualise the data in form of ground track plots, time period plots and simple indicators.

3. Example diagram – determination of satellite trajectory

When artificial satellite orbits the Earth it is important for the ground controllers to be able to predict its future position. This is necessary to plan the communication sessions and scientific or commercial operations of the satellite payload. Spacecraft position very often needs to be represented in terms of Mercator projection of satellite trajectory on a world map and its attitude above the Earth. The reason is that ground stations, and observation targets rotate together with the Earth. One of the ways of representing satellite orbital position are so called Keplerian elements. Those parameters describe shape of the orbit, and position of the spacecraft. Due to various effect (most importantly Earth oblateness) orbit shape and orientation fluctuates over time compared to trajectory around an ideally spherical body. There are various way to measure current orbital parameters of the satellite. In most cases it can be obtained from North American Aerospace Defence Command (NORAD)[10]. Information of satellite orbit available at certain time can be used to predict future orbital parameters with orbit propagators. Diagram shown in Figure 3 allows to better understand how such a prediction can be obtained.

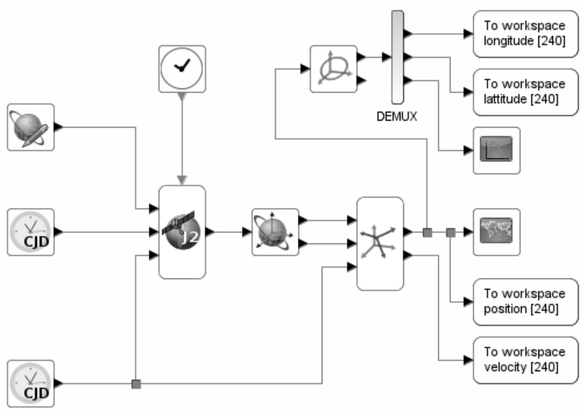

Fig. 3. Xcos diagram allowing for satellite trajectory determination

AB implements three orbit propagators to be chosen from. They offer a various level of precision starting from simple Keplerian propagator, through J2 propagator [6] that takes into account Earth oblateness and ending with the sophisticated Brouwer-Lyddane propagator [4]. In this example J2 propagator is used, and represented on the diagram in Figure 3 by the rectangular block with icon containing 'J2' caption. Block has three inputs. First one from the top comes from the block allowing for input of the initial Keplerian elements of the orbit. This user input is validated, to make sure that only values from allowable range are fed to the algorithm. Second input allows entering the Julian day time representation [7], which is often used in aerospace algorithms. The third input allows to provide the time for which the satellite position should be propagated. During the simulation time is progressing in a preconfigured way to allow for calculation of the satellite trajectory. The J2 propagator block has a single output, that represents the Keplerian orbit parameters of the satellite at the current simulation time. Obviously, this representation of satellite position is not convenient for interpretation, therefore a Cartesian conversion block is used to retrieve position (first output) and velocity (second output) in the internal Cartesian frame originating at Earth centre. This operation is still insufficient to localize the satellite with respect to the ground station because it is necessary to take into account the fact that Earth is at the same time revolving around its own axis. Frame conversion block is used to convert between inertial, and non-inertial Cartesian frame tied to the Earth. Note that for this operation it is again necessary to feed the block with current simulation time. Information about satellite position and velocity relative to the Earth can be then saved as numerical data by utilizing 'To workspace' blocks. It is also possible to use 'Plot groundtrack' block that interprets it geographically and allows plotting the satellite trajectory, as shown in Figure 4.

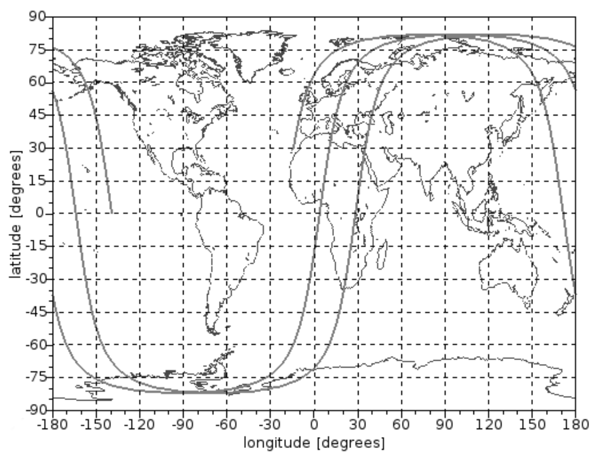

Fig. 4. Plot showing satellite groundtrack trajectory

4. Example diagram – determination of ground station access times

If satellite ground controllers need the information about time periods when the satellite will be visible (direct line of sight also allows radio communication) from the ground station located in specific point on the globe it is necessary to perform some additional calculations.

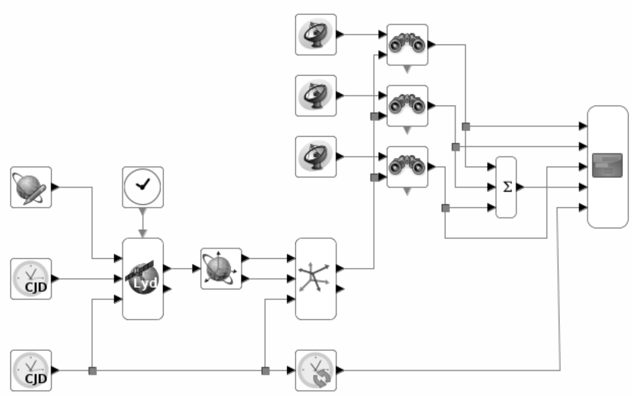

Fig. 5. AB diagram solving problem of satellite ground station access time calculation

Diagram in Figure 5 has some additions in comparison with Figure 3. First of all it is worth noting that it uses a more precise Lyddane orbit propagator [4]. Secondly, three separate ground stations are defined with block depicted by satellite dish. This is done by defining ground station names, their geographical longitude and latitude, elevation above the sea level and the mask value. Mask is an angle value given in degrees defining the minimal elevation of the satellite above the horizon that allows for existence of the direct line of sight. This enables taking into account any obstacles on the horizon that may interfere with the communication.

This information, together with satellite position can be then fed to the AB blocks that determine if the direct line of sight exists. Those are marked by an icon with a pair of binoculars. Plotting block located on the right hand side of the diagram uses this information to draw a visibility plot. Note that the summational block is used to create additional visibility signal determining if any one of the ground station sees the satellite. Time is also fed to the plot (after being converted to the hh:mm format) to make it possible to label beginning and end of each of the visibility periods. Figure 6 produced by the example diagram shows that in this particular case first ground station sees the satellite between 3:35 AM and 3:45 AM, second one between 3:40 AM and 3:50 AM, and the third one between 3.59 AM and 4:09 AM. Those short visibility periods are typical for Low Earth Orbit satellites, and require accurate communication window predictions.

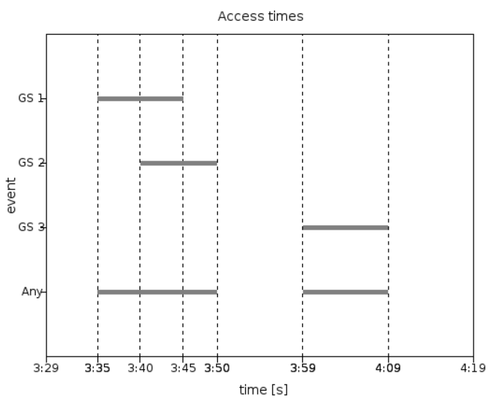

Fig. 6. Example plot showing satellite ground station access times

5. Example diagram – position and motion of heavenly bodies

It is also possible to simulate spacial position of Heavenly bodies such as planets, the Moon, and the Sun. Figure 7 depicts trajectories of inner Solar system planets generated by another demonstration diagram of the AB.

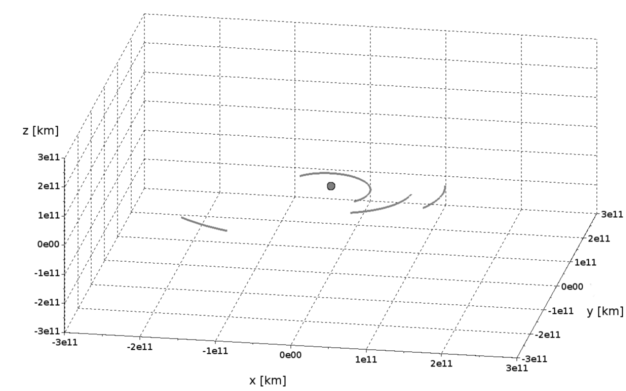

Fig. 7. Trajectories of the Mercury, the Venus, the Earth and the Mars (in order of distance from the Sun)

6. Example diagram – environmental models

Apart from AB blocks Xcos itself also contains wide range of other functions. They can be used in conjunction to build even more sophisticated diagrams. Demonstration shown in Figure 8 is an example of such a hybrid. In this scenario it is assumed that stratospheric meteorological balloon bursts at the attitude of 24 km. The electronic equipment enclosed in a capsule of mass equal to 1.85 kg then descends on a parachute. AB atmospheric model block, unit conversions and distance calculations blocks are used in conjunction with standard Xcos blocks to solve the differential equation of motion.

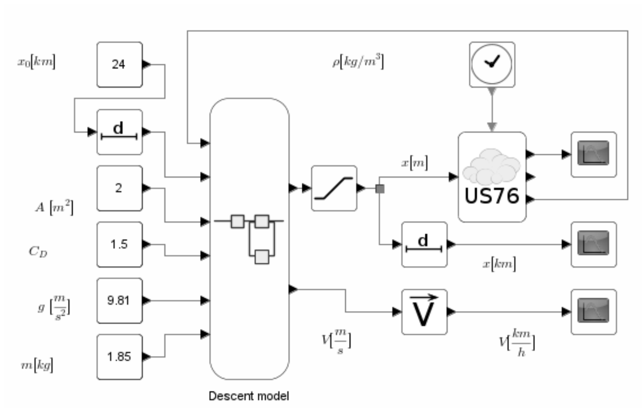

Fig. 8. Xcos diagram used to calculate parachute descent profile

This allows to estimate the altitude of the craft over time, and predict the time needed to land and speed at which it will touch the ground. Plot of altitude versus time produced by this model is shown in Figure 9. The descent time in this example turns out to be around 70 min.

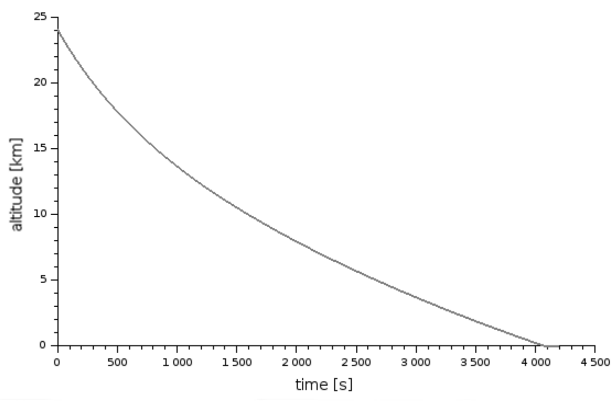

Fig. 9. Example plot showing weather balloon capsule descent profile

7. Example diagram – spacial orientation algebra

Algebraic functions provided in AB mainly focus on enabling one of the most important type of aerospace calculations, namely the attitude. Euler angle and axis, direction cosine matrix, and attitude quaternions are some of the most popular attitude representations. Understanding relationships between them and performing all essential mathematical operations is also possible with the blockset.

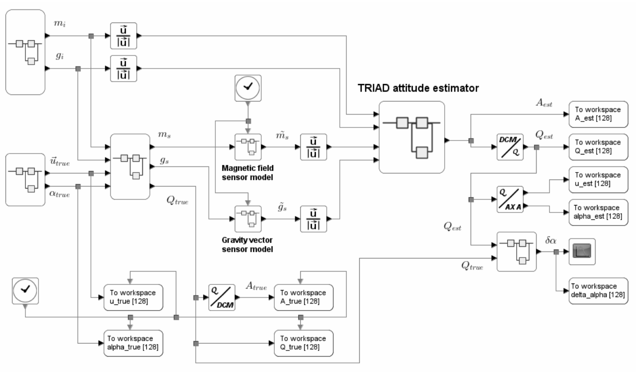

Fig. 10. Diagram testing performance of the TRIAD attitude estimation algorithm

Some of those are demonstrated in the example diagram shown in Figure 10. A well known TRIAD attitude estimator is constructed from elementary blocks. Attitude sensors are simulated with simplistic noise models and the precision of attitude estimation is tested.

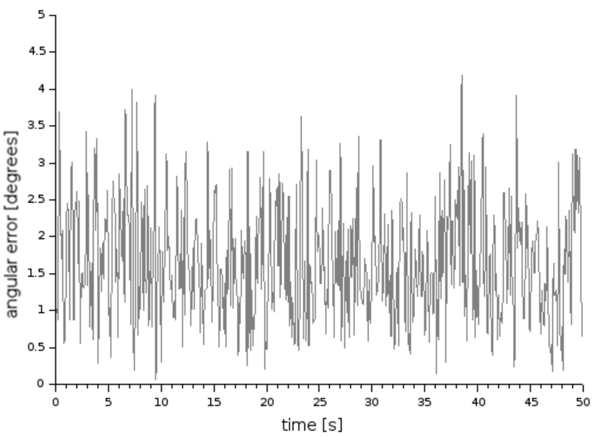

Fig. 11. Example plot showing angular error of attitude estimation

Plot in Figure 11 shows the angular error of attitude estimation in degrees over the simulation period.

8. Summary

Accession of Poland to the European Space Agency stimulated increased interest in space-related research and education. Aerospace Blockset, the software tool described in the paper supports developing solutions to aerospace problems relatively intuitively. This has a great potential to enhance research, design, and educational capabilities in this field. Being free it also allows for such an education to be conducted in any institution without the need of acquiring additional, typically expensive software. However, because it is being based on a professional space mechanic library it can be utilized in real satellite mission design and operation.

It is important to note that the AB project was co-funded by the European Space Agency as part of Summer of Code in space initiative.

9. References

[1] Scilab Enterprises.: Scilab: Free and Open Source software for numerical computation. Orsay, France. 2012.
[2] Wang L., et al.: Active Disturbance Rejection Control simulation toolbox in open-source software Scilab/Xcos. 2011 International Workshop on Open-Source Software for Scientific Computation. IEEE, 2011.
[3] Leros A., and Andreatos A.: Using Xcos as a teaching tool in simulation course. Proceedings of the 6th International Conference on Communications and Information Technology (CIT'12), 2012.
[4] Lyddane R. H.: Small eccentricities or inclinations in the Brouwer theory of the artificial satellite. Astronomical Journal 68, 555–558, 1963.
[5] National Aeronautic and Space Agency.: U.S. Standard Atmosphere. 1976.
[6] Neta B.: Partial List of Orbit Propagators. http://calhoun.nps.edu /bitstream/handle/10945/39471/list.pdf?sequence=1 2005.
[7] Meeus J.: Astronomical Algorithms. Willmann-Bell 1991.
[8] Finlay C. C., Maus S. et al.: Evaluation of candidate geomagnetic field models for IGRF-11. Earth Planets Space, 2010.
[9] Heidt Hank, et al.: CubeSat: A new generation of picosatellite for education and industry low-cost space experimentation, 2000.
[10] http://www.norad.mil/

Received: 11.07.2015 *Paper reviewed* *Accepted: 03.11.2015*

Paweł ZAGÓRSKI, MSc, eng

Teaching Associate at AGH University of Science and Technology in Kraków. Author of scientific publications awarded with several scholarships from European Space Agency. His interests include state estimation, satellite attitude determination and space systems engineering.

e-mail: pzagor@agh.edu.pl

Prof. Wojciech GREGA, PhD, eng

Professor at AGH University of Science and Technology in Kraków. Author of over 150 scientific publications including three books. His work concentrates on digital control, distributed control and modeling and control of energy production and distribution systems. He also published a number of articles on teaching methodology in higher education.

e-mail: wgr@agh.edu.pl