v. 4, n. 13, 2024

Journal of
Engineering
Research

USER INTERFACE DEVELOPMENT FOR THE PLATO SCIENTIFIC SATELLITE ATTITUDE CONTROL SYSTEM SIMULATOR

Lucas Wu Jiajun
Department of Telecommunications and Control Engineering
``Escola Politécnica da Universidade de São Paulo``

Fábio de Oliveira Fialho
Department of Telecommunications and Control Engineering
``Escola Politécnica da Universidade de São Paulo``

DOI https://doi.org/10.22533/at.ed.3174132429044

Abstract: PLATO (PLanetary Transits and Oscillations of stars) [1] is a European space mission with Brazilian collaboration focused on the search for exoplanets of the typeterrestrialin habitable zones of stars similar to the Sun. Conducted by the European Space Agency (ESA), it is scheduled for launch in 2026. One of the main products of the mission are the light curves of a large number of stars. The method used, fixed aperture photometry, aims maximized signal-to-noise ratio of light curves. For this purpose, several noises are considered, including those from the CCD detector (Charge-Coupled Device), zodiacal light and satellite movement (jitter). A method Compensation algorithm based on CoRoT satellite (CNES) algorithms was developed to correct photometric jitter noise. For the correct implementation of the method, it is necessary to use an Attitude Control System simulator. Such a simulator is under development within the scope of research dedicated to PLATO. It is in this context that the project, whose objectives are the development of a graphical user interface and its integration into the PLATO mission Attitude Control System simulator. It is expected with this work that the use of the simulator within the consortium of the mission can be carried out by non-control specialists, and in support of the development of many different mission instrumental correction algorithms.

Keywords: Dynamic simulation. Attitude control system. Instrumental corrections. PLATO satellite.

INTRODUCTION

The Automation and Control Laboratory (LAC/PTC) participates in the PLATO mission (ESA) providing algorithms for instrumental corrections of scientific data. Among them is the satellite jitter correction algorithm[2], which requires the development of a simulator of the satellite's ACS (Attitude Control System). This development has been underway since 2020.

At the same time, LAC also developed an ADCS (Attitude Determination and *Control System) for CubeSats, with validation using flight data from the PicSat mission (Pictoris Satellite), from the Paris-Meudon Observatory, France [*3]. Based on this simulator, LAC is currently cooperating with the Paris Observatory in the development of CubeSat CASSTOR, a scientific mission dedicated to the observation of stellar magnetospheres that will serve as a technological validator. An expanded version of the simulator is being developed, including interface user interface and simulation and flight data analysis tools.

Therefore, this project consists of developing data analysis tools for the CASSTOR satellite. And, later, integrate them into the PLATO mission, enabling the exploration of synergies between both simulators, reducing efforts and optimizing the techno-scientific return of the development of these tools. Such tools, carried out with the help of MATLAB software (MAtrix Laboratory), in addition to its extensions, consist of human-machine interaction programs with graphical interfaces integrated into the PLATO simulator to be used by the different countries participating in the mission.

OBJECTIVES

The objectives of this project are the development of a graphical user interface and its integration into the PLATO mission Attitude Control System simulator. The human-machine interaction tool, developed in the App Designer of MATLAB aims to help the team in charge of the PLATO mission.

This tool is designed to offer an intuitive interface, considering the presence of team members who do not have expertise in control systems. Furthermore, the aim is to provide an agile response to user interactions, an essential factor for efficient mission operations.

Integration with the ADCS simulator constitutes a crucial step towards validating this system. This integration not only makes it easier to carry out tests and verify the simulator's performance using PicSat data, but it also proves to be very useful during mission operation. The ability to generate predictive information about the position and orientation of the satellite at future times, or during intervals in which telemetry data is not available, is critical to the efficiency and success of the mission. Therefore, effective integration as an ADCS simulator is an essential component to ensure reliability and accuracy in satellite operations throughout the entire mission.

METHODS AND PROCEDURES

The development of this project is carried out with the help of MATLAB software, including its Simulink tools for dynamic modeling and App Designer for creating the user interface. The hierarchical organization and structuring of the data to be analyzed is carried out using the HDF View software, which allows the visualization and editing of data stored in the HDF5 (Hierarchical Data Format 5) format, a high-quality data structure. performance, simple handling and easy sharing.

The data provided by the PicSat mission was employees to test all the functionalities of the human-machine interface.

The project was divided into the following stages:

- Initial prototype of the graphical interface with flight data files from PicSat in Format FITS (Flexible Image Transport System);
- Creation and structuring of the data file in HDF5 format, the format used by the simulator, from FITS files;
- Reading and analyzing results using a graph visualization application interactive, making it possible to compare telemetry and simulation;
- Integration of interface data with the simulator, performing the data flow between the tools allowing to change the satellite's operating modes in the simulator using the user tool;
- Optimization of interface processing with the aim of ensuring operational efficiency and improving the user experience;
- Comparison of simulation results with telemetry data from PicSat, making it possible to verify the accuracy and validity of the simulator.

RESULTS

In the interface, we implemented functionalities that enable graphical visualization of data obtained from an HDF5 file [figure 1]. These data incorporate information from the PicSat satellite, such as measurements in the three axes of attitude, angular velocity and magnetic field, in addition to the currents related to sensors, actuators and the ADCS system. Additionally, you can overlay the telemetry graph with the data imported from the simulator output to compare them.

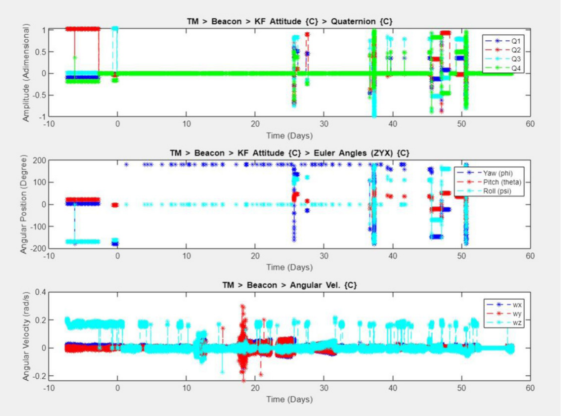


Figure 1: Graphical interface window in operation

Another incorporated tool was the visualization of commands sent to the satellite. Thus, it is possible to verify the operation of control modes and actuators through orientation and angular velocity measurements. However, the PicSat command signals were sent using hexadecimal instructions, making it necessary to use the documentation of the onboard ADCS. By analyzing and comparing the graphs below [figure 1], a correlation is identified between the satellite's angular velocity and the activation of the Detumbling command.

However, this observation alone is not conclusive, considering the presence of other operating modes that also influence the satellite's behavior [figure 3].

Regarding high-level commands, instructions are initially formulated in a vector based on decimal values, which are later converted to hexadecimal format in order to be used by the satellite.

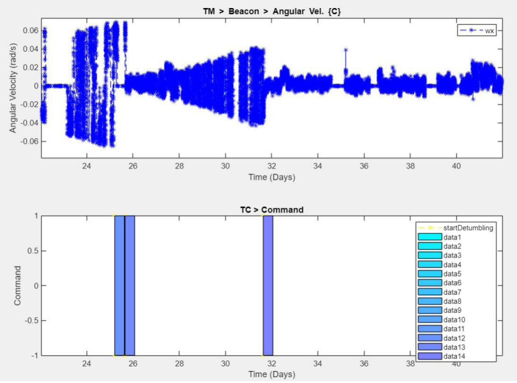


Figure 2: Plots of angular velocity in the x-axis of detumbling mode over time

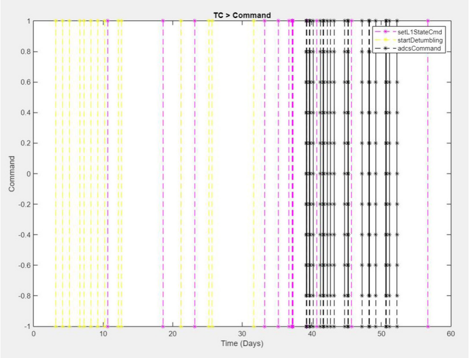


Figure 3: Chart of commands performed by PicSat

This conversion covers various essential information, such as time, calibration procedures, actuator references, among other relevant data. Therefore, it is necessary to understand why complete the meaning of each hexadecimal instruction to enable implementation with the simulator.

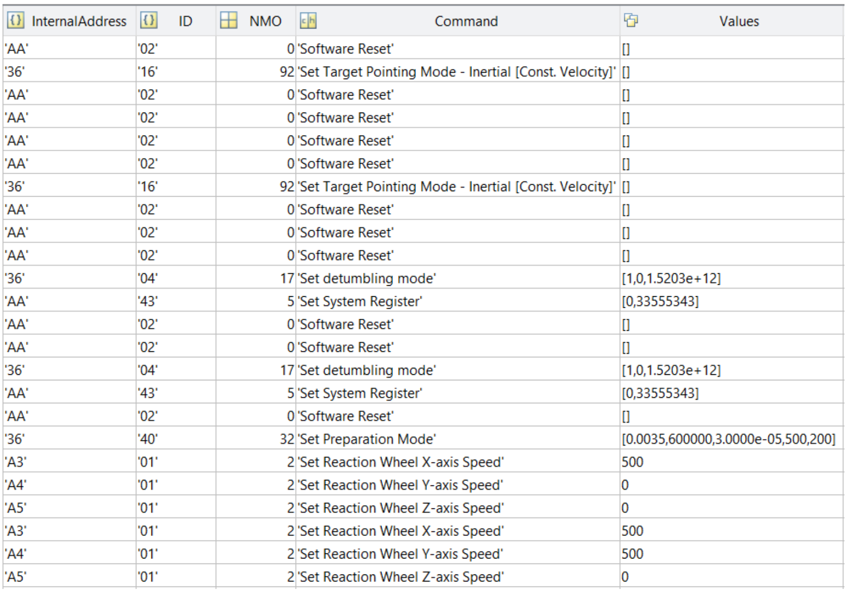
Instructions are sent by the OBC (On-Board Computer) to the iADCS, these are made up of hexadecimal numbers ranging from 0x00 to 0xFF, indicate that they are formatted in bytes. A convention in the storage edge used by the responsible company was big-endian, that is, with the most significant bit on the left.

A set of one or more bytes determines the type of instruction or the values associated with that instruction, varying according to the format: byte, word (2 bytes), integer (4 bytes), long integer (8 bytes), single (4 bytes) or double (8 bytes), the last two being floating point representations according to the IEEE 754 standard.

The number of bytes required for each information depends on what its value represents, for example internal addresses, number of OBC outputs (ADCS inputs), command types and numerical values that can represent the duration of the command as well as physical quantities of actuators and sensors.

This way, it becomes possible to analyze which commands were executed by the satellite throughout its mission, examining specific sequences of instructions transmitted from the OBC to the ADCS and translating them to enable human interpretation in the graphical tool [figure 4].

Thus, the user has the ability to select a specific interval of execution of a high-level command, and the pertinent information, such as new initial conditions and a new operating mode are sent to the simulator. This process allows you to recreate or predict the trajectory and states of the satellite. This



InternalAddress	ID	NMO	Command	Values
'AA'	'02'	0	'Software Reset'	[]
'36'	'16'	92	'Set Target Pointing Mode - Inertial [Const. Velocity]'	[]
'AA'	'02'	0	'Software Reset'	[]
'AA'	'02'	0	'Software Reset'	[]
'AA'	'02'	0	'Software Reset'	[]
'AA'	'02'	0	'Software Reset'	[]
'36'	'16'	92	'Set Target Pointing Mode - Inertial [Const. Velocity]'	[]
'AA'	'02'	0	'Software Reset'	[]
'AA'	'02'	0	'Software Reset'	[]
'AA'	'02'	0	'Software Reset'	[]
'36'	'04'	17	'Set detumbling mode'	[1,0,1.5203e+12]
'AA'	'43'	5	'Set System Register'	[0,33555343]
'AA'	'02'	0	'Software Reset'	[]
'AA'	'02'	0	'Software Reset'	[]
'36'	'04'	17	'Set detumbling mode'	[1,0,1.5203e+12]
'AA'	'43'	5	'Set System Register'	[0,33555343]
'AA'	'02'	0	'Software Reset'	[]
'36'	'40'	32	'Set Preparation Mode'	[0.0035,600000,3.0000e-05,500,200]
'A3'	'01'	2	'Set Reaction Wheel X-axis Speed'	500
'A4'	'01'	2	'Set Reaction Wheel Y-axis Speed'	0
'A5'	'01'	2	'Set Reaction Wheel Z-axis Speed'	0
'A3'	'01'	2	'Set Reaction Wheel X-axis Speed'	500
'A4'	'01'	2	'Set Reaction Wheel Y-axis Speed'	500
'A5'	'01'	2	'Set Reaction Wheel Z-axis Speed'	0

Figure 4: Translation of PicSat command instructions

functionality enables integration between the tools involved in the project, facilitating analysis and visualization of the effects of high-level commands on the satellite's behavior. Therefore, it is necessary export data from the graphical interface to the simulator, a task which is facilitated by the HDF5 file format and its support in MATLAB.

CONCLUSIONS

The graph generation tools in application were effectively implemented, as well as applications and optimized for the experience of user. From this, it becomes possible to integrate this system with the simulator to validate the preliminary results obtained.

Following this integration, the reconstruction of the satellite's trajectory, a necessity due from occasional loss of flight data. This crucial phase will allow for detailed comparative analyzes between simulated results and telemetry data.

THANKS

The authors would like to thank FAPESP, which finances Brazilian participation in PLATO via thematic, process number 2016/13750-6, including the present work with scientific initiation scholarship with process number 2023/10763-3.

In addition to thanks to Cauê Garcia Menegaldo, Karoline Carvalho Bürguer and Luca Augusto Paniago for their assistance in developing the work.

REFERENCES

[1] H. Rauer et al. "The PLATO 2.0 mission". Em: *Experimental Astronomy* 38.1–2 (set. de 2014), pp. 249–330. ISSN: 1572-9508. DOI: 10 . 1007 / s10686-014-9383-4.

[2] R Drummond et al. "Jitter Correction Algorithms for the COROT Satellite Mission". Em: Publ. Astron. Soc. Pac. 118.844 (jun. de 2006), pp. 874–884.

[3] Cauê G. Menegaldo et al. "PicSat's Enduring Legacy. Probing the Flight of a Small Astronomical Satellite". Em: Publications of the Astronomical Society of the Pacific 134.1033 (mar. de 2022),p. 034501. DOI: 10.1088/1538-3873/ac4e1d.

[4] Cauê Garcia Menegaldo. "Simulador de controle de atitude e propagação de órbita aplicado a nanossatélites em órbita baixa terrestre: desenvolvimento e validação com dados de voo do nanossatélite PicSat.." Em: *Dissertação (Mestrado em Engenharia de Sistemas) - Escola Politécnica, Universidade de São Paulo* (2020). DOI: 10.11606/ D.3.2020.tde-09032021-083208.

[5] Matthias Buhl. *Interface Control Document*. Berlin Space Technologies, 2018.