v. 4, n. 12, 2024

Journal of
Engineering
Research

DEVELOPMENT OF GRAPHIC INTERFACE FOR THE CASSTOR SCIENTIFIC SATELLITE ATTITUDE CONTROL SYSTEM SIMULATOR

Luca Augusto Paniago
Department of Telecommunications and Control Engineering, Electrical Engineering Building, ``Escola Politécnica da Universidade de São Paulo``

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

Abstract: This article aims to present the development of a Human-Machine Interface (HMI) for the dynamic ADCS (Attitude Determination and Control System) simulator for nanosatellites, developed by LAC/EPUSP, and validated with flight data from the PicSat satellite (Observatory of Paris), to be applied in the CASSTOR mission, in partnership with the Paris Observatory. This way, a user with little experience in ADCS can perform dynamic simulations with the help of the interface, in which the parameters of the satellite in question, disturbances, initial conditions, control modes are inserted, in addition to viewing the results of the simulations to validate the subsystem design. In this interface, all the functionalities of the PicSat mission, adopted in the dynamic simulator, were integrated. And new features related to the CASSTOR mission are being added.
Keywords: Nanosatellites, ADCS, Simulation, PicSat, CASSTOR.

INTRODUCTION

Since the advent of CubeSats, satellites made up of cube-shaped modules, with an edge of 10 cm and no more than 1.33 kg per cube, which occurred in 1999, there has been a great expansion in the use of this equipment. In 2024, they areexpected576nanosatellite launches (https://www.nanosats.eu/img/fig/Nanosats_years_black_2022-08-01.png), class of which cubesats form the vast majority. Up to 01/2024, adding up every year, of the 2532nanosatellites launched,2323they were CubeSats. It is, therefore, a new paradigm in the space area and a volume of launches that exceeds the launch of conventional satellites, that is, medium and large satellites, by approximately one order of magnitude.

Interest in the technological domain of these devices continues to grow, as do their market and applications.

A critical and strategic technology in satellites is the ADCS (Attitude Determination and Control System). This system provides the satellite with the ability to point in three-dimensional space, allowing its mission to be carried out.

In 2013 there was a boom in the use of CubeSats. It was the year in which the first CubeSats controlled in their three rotation axes using stellar sensors were launched. Since then, applications in terrestrial imaging and scientific observations have become possible.

The Automation and Control Laboratory (LAC) of USP Polytechnic School has been working on the development of technologies necessary for ADCS since 2014, having developed in particular an ADCS simulator (Menegaldo et al. 2022 and Menegaldo 2020), with validation using flight data from the PicSat mission, from the Paris Observatory, France. This is the first validated Brazilian ADCS simulator.

Currently, the Observatory from Paris works on the development of CubeSat CASSTOR, a scientific mission dedicated to observing stellar magnetosphere sat the tenth of a second of bow. In this context, LAC-EPUSP will complete the development of the ADCS simulator so that it can serve as a support tool.

Soon, it is necessary to develop a graphical interface that simplifies the use of this simulator in the context of the CASSTOR mission, which will speed up the mission development process and eliminate the need for users with too specific training in the Control area to operate it.

The structure of the simulator in SimuLink is shown in the figure below:

Therefore, the objectives of this work are the development of a graphical interface and its integration with the ADCS simulator of the CASSTOR mission.

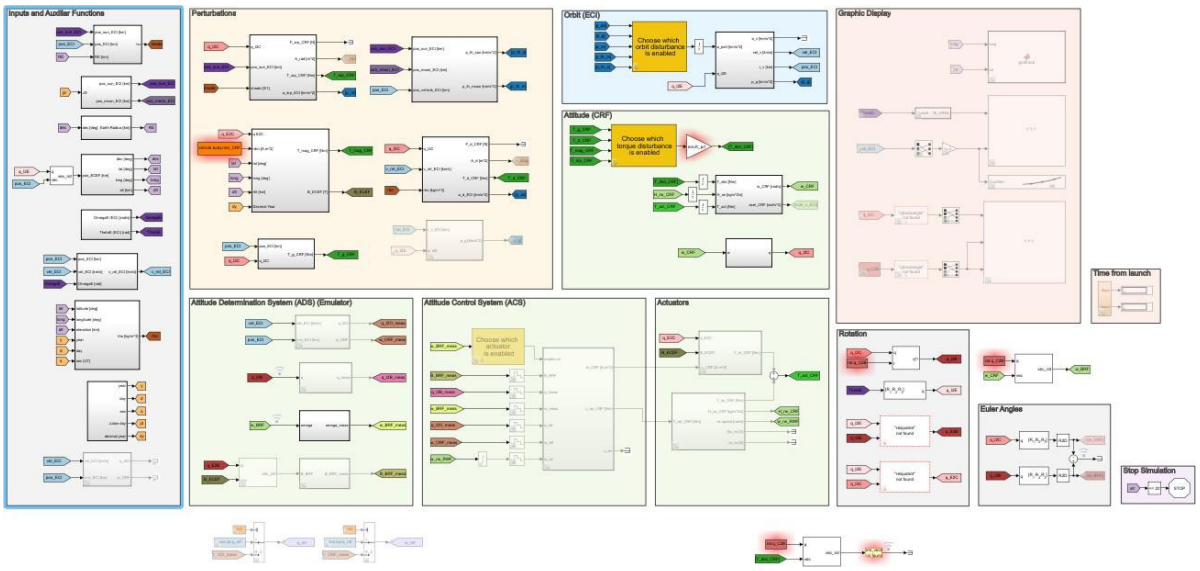

Figure 1 – ADCS simulator in Simulink.

METHODOLOGY

This development is being done with the help of the MATLAB/Simulink computational tool. In particular, the App Designer tool. This tool allows the interactive development of the interface from the library of blocks and functions, such as texts, tabs, numeric editable ports, graphics, buttons, etc. Furthermore, the interface can be created using the MATLAB code language, configuring "Callback Functions" of the created blocks, such as, for example, the system's response to clicking a certain interface button, as well as the interface design itself, among other things. others.

Below the HMI (InterfaceHuman-Machine) there are a number of functions partially developed to connect its content to that of the ADCS simulator, presented in the Results section.

The entire interface between HMI and Simulink will be done via HDF5 files (Hierarchical Data Format 5), high-performance data structure, simple to handle and share, so that the simulator can be used without the need for HMI intermediation. To the first and the project's milestones are the organization of the input data for the simulator in this format, and the development of an algorithm for reading and converting this data to the Matlab environment.

The next step in the methodology is to design the interface, arranging the tabs, simulation data and result plots conveniently for the best user operation, seeking to be intuitive and efficient.

With the design completed, it is necessary to adapt the variables created in the App Designer environment to those used in the simulator in its scripts and in the block diagram in Simulink, so that the HDF5 inputs are sent to the simulator, as well as the simulated results are interpreted. and presented on the interface. At this stage, functional tests were carried out, comparing what was expected by the simulator with what is actually presented in the HMI, in addition, due to the volume of data, they were divided by functionality, without the need for analysis and validation of the dynamics and methods. control used for now.

RESULTS

HDF5 DATA STRUCTURE

This section presents the organization of the HDF5 input data used in the developed ADCS simulator. We sought to create the data structure within the file following the same logic as the data presented in the interface (Figure 4), for example, body data in "Body", sensors in "Sensors", etc. Furthermore, this division was used in the variables used in Matlab, through the implemented Object-Oriented Programming (OOP), so that they were encapsulated in the same "data" object. For example, the variable referring to the engine mass is recorded as: "data. Motor. Mass". Figure 2 presents this data in the HDF View application, in which the structure was created.

The "Description" and "Units" tabs were added, in addition to the numerical values of the fields, so that, ultimately, the end user has a better usability experience.

An HDF5 "Default" file was created containing the PicSat satellite parameters to be used as the user's "starting point" when starting a new simulation. This way, the user does not have to fill out all the dozens of interface fields in full. Simply update the specific fields for the simulation you want to perform, saving this new configuration in a new HDF5 file. This functionality was implemented through the "New" functions, in which when called the interface imports the standard PicSat parameters, and also through the "Save" and "Save as" functions, in which the user registers this new configuration. Access to these functions through the interface is shown in Figure 3.

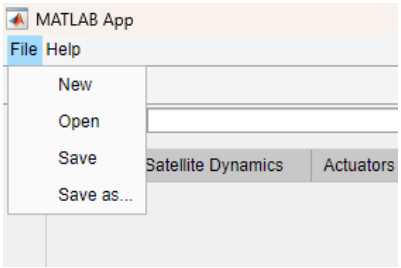

Figure 3. Organization of simulator input parameters by HDF5

HMI DESIGN

The logic of the partially developed interface is the division of work areas into tabs in a similar way to the organization of the HDF5 file presented above. So, in each tab there are possible input parameters for the simulation, which the user can change directly, which are in fact related to the name of that tab. Furthermore, plots of interest from the simulator can be shown. When accessing the interface, whose initial tab is shown in Figure 5, the data file whose directory is shown in "Filename" is automatically read. If you wish to load another file, you can select the item shown below "File". If any changes were made to the parameters directly from the interface, the icon just below "Help" can be selected to save them, updating the HDF5 file. Figures 4, 5, 6 and 7 show the design of some tabs with their respective examples of simulation parameters.

INERTIA ESTIMATION VIA HMI

When developing the simulator, the input parameter of the body's Inertia Tensor is required to calculate the satellite's dynamics. In fact, when simulating the attitude control of his mission, the user can have this value estimated via CAD, for example, since the project is in an advanced stage of development. However, this is often not the case, so a functionality was implemented so that the user has the option of individually setting the inertia parameters of each satellite subsystem. This information

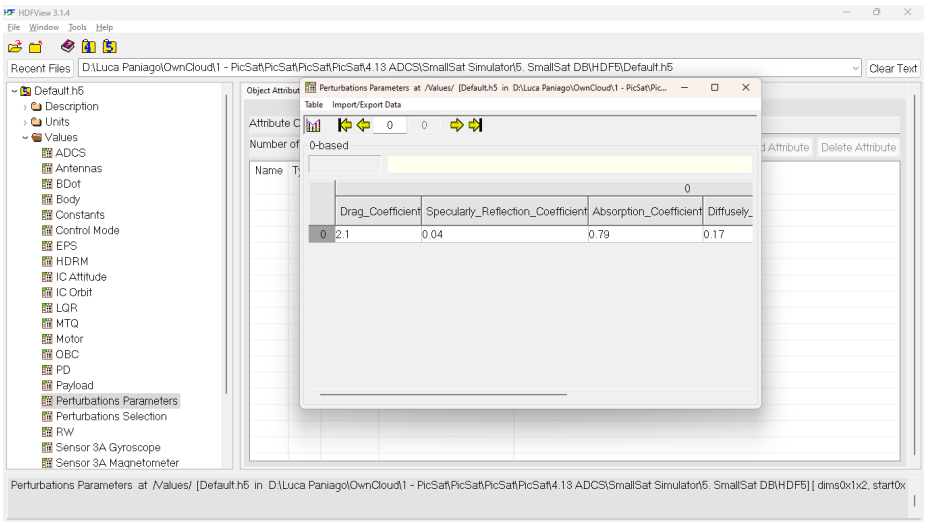

Figure 2. Organization of simulator input parameters by HDF5

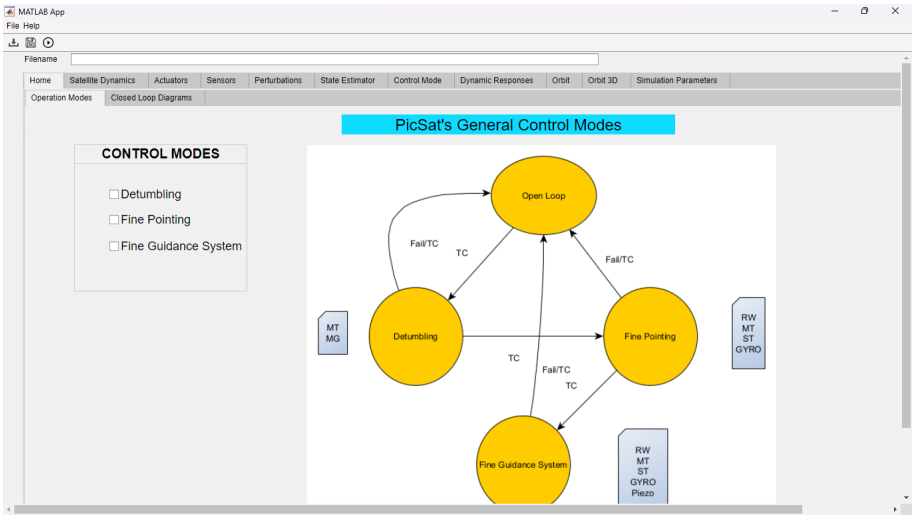

Figure 4. Graphical interface home page

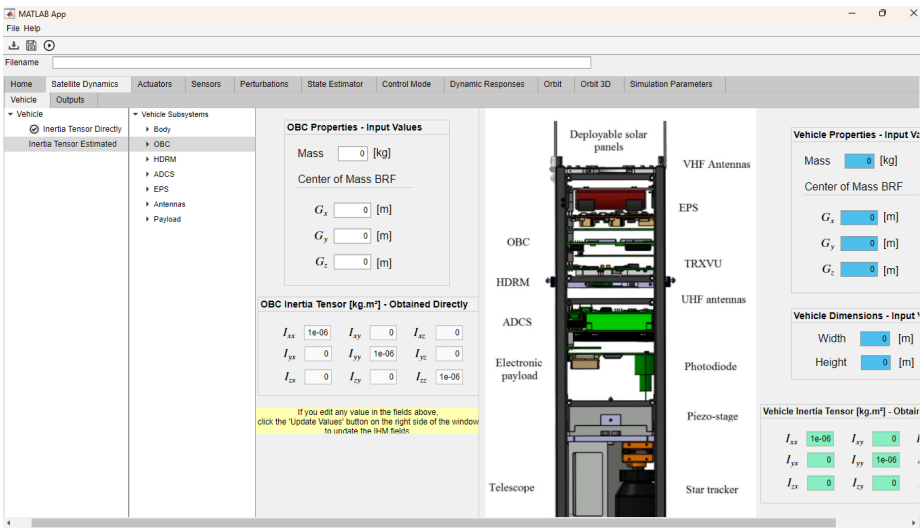

Figure 5. HMI “Satellite Dynamics” tab

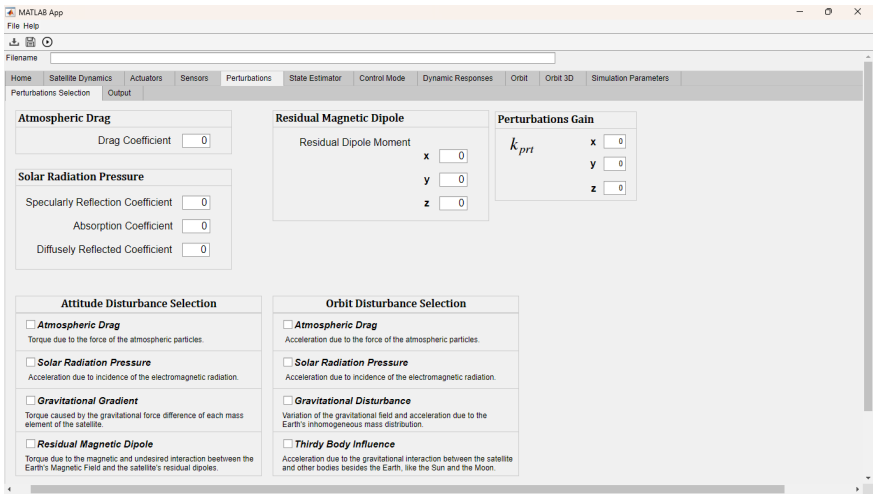

Figure 6. IHM "Perturbations" tab

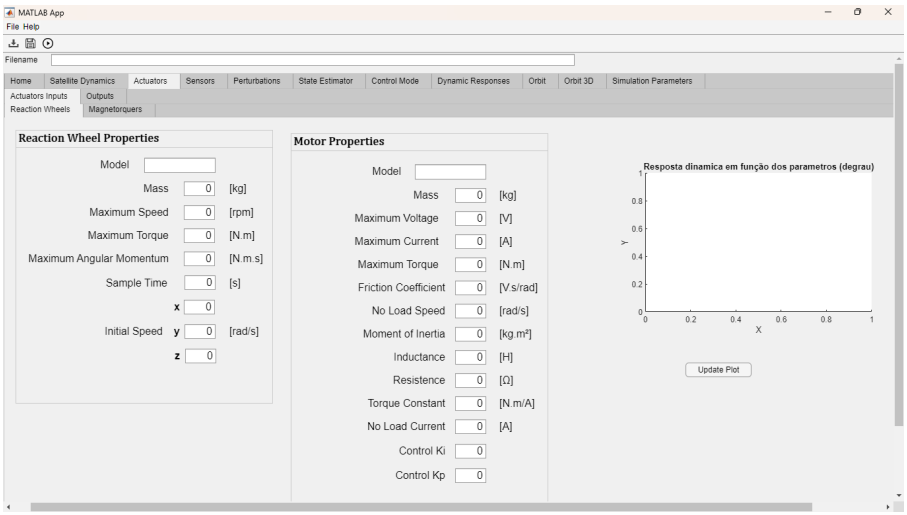

Figure 7. HMI "Actuators" tab

from the subsystems, therefore, can also be imposed via the Inertia Tensor directly, or through dimension and mass data, which the HMI calculates the resulting inertia tensor. Finally, all the subsystem tensors are added together, composing the vehicle's final tensor, which is what is really necessary for its dynamics.

Figure 8 depicts the case in which the user just adds the satellite's final inertia tensor, since this data is already calculated externally. In Figure 9, the option to calculate the inertia tensor using subsystem data is shown, directly inserting the respective inertia tensors. Finally, Figure 10 presents the case of estimating the inertia tensor of the subsystems, and consequently of the satellite.

Furthermore, each subsystem has a "summary tab", which provides the current center of mass, mass and inertia tensor of each of them. This tab is exemplified in Figure 11 with the "Body" subsystem, linked to the CubeSat structure. In any case, regardless of the option adopted by the user regarding the vehicle's inertia tensor, to the right of the main "Vehicle" tab there is also information on the inertia tensor, center of mass, mass and dimensions of the satellite as a whole, to facilitate possible user interactions. If any change is made to any parameter on this

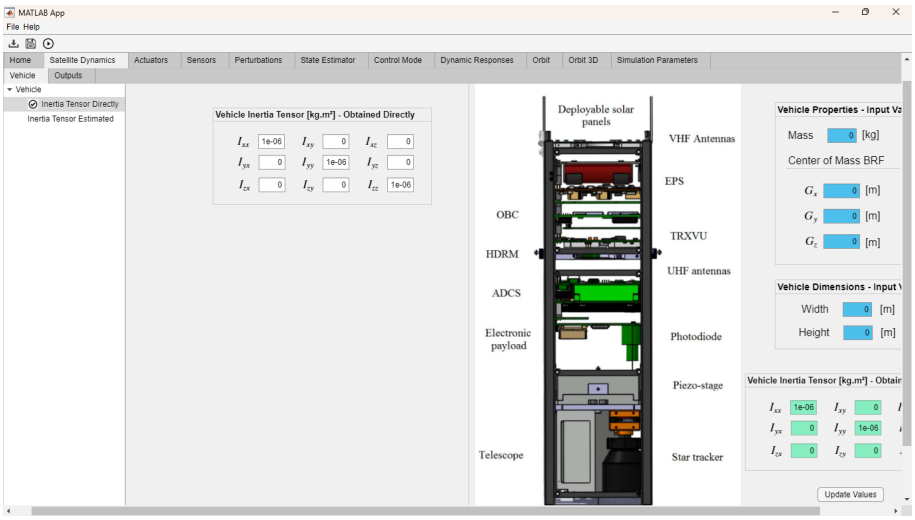

Figure 8. Vehicle inertia tensor option provided directly by the user

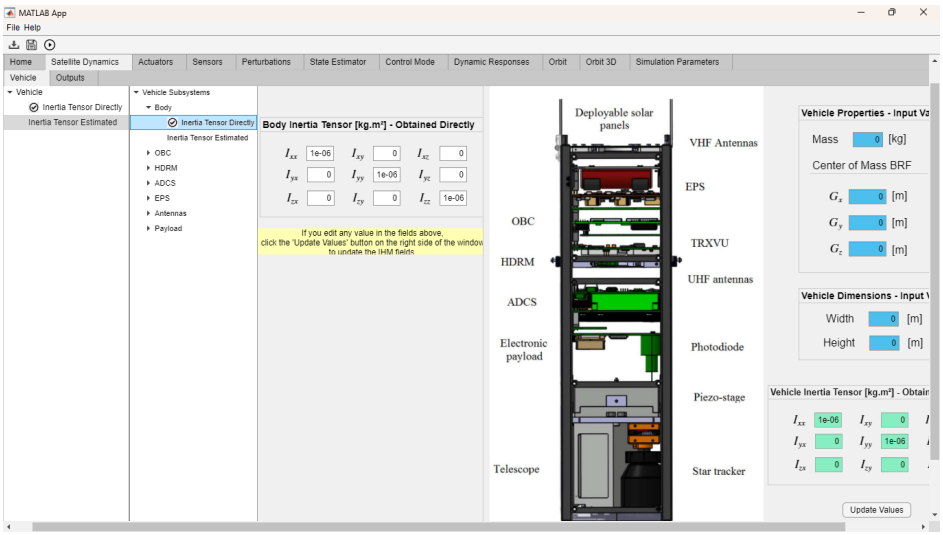

Figure 9. Vehicle inertia tensor option estimated via user-supplied subsystem inertia tensors

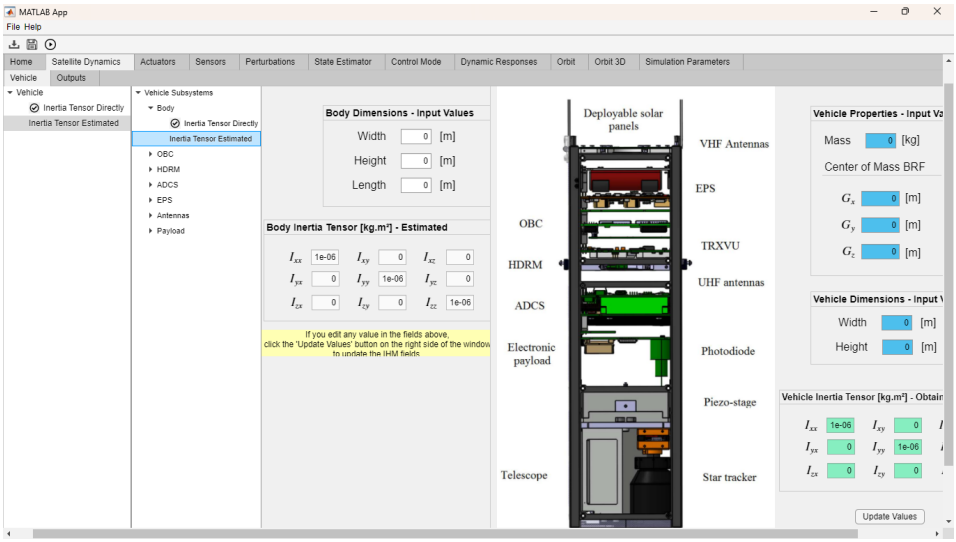

Figure 10. Inertia tensor option for the vehicle and subsystems estimated by the HMI

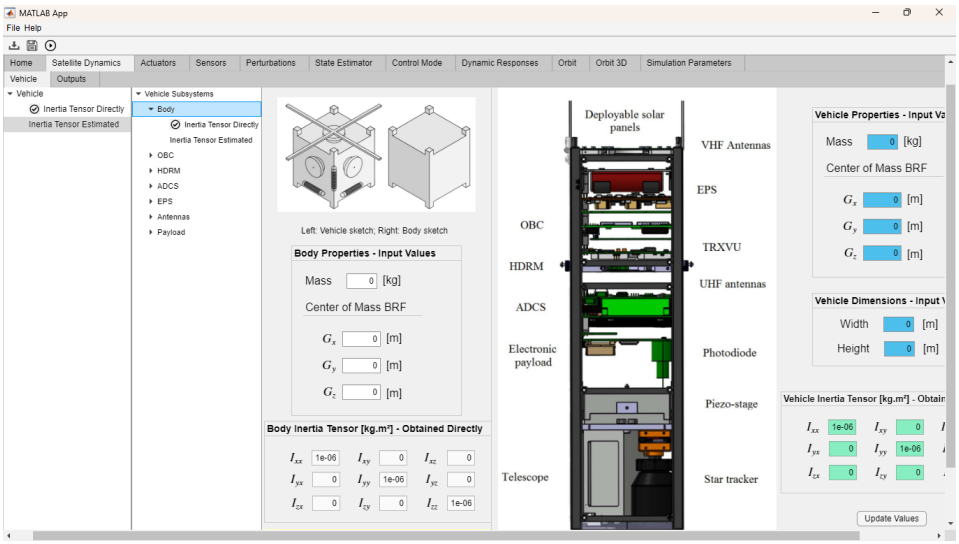

Figure 11. "Body" subsystem summary tab

tab, to update the vehicle's final information, simply press the "Update Values" button in the bottom right corner of the tab.

HMI SIMULATION RESULTS

To carry out the simulation, simply select the "Play" icon, next to the icon to save the simulation parameters. The simulator scripts will be read and Simulink opened. After the simulation, the results are presented in the respective interface tabs, as shown in Figures 12 and 13.

ORBIT SIMULATION

With the aim of enabling the user to carry out a more detailed and extensive analysis of their satellite's orbit in a reduced simulation time, a simplified simulator was used, with only orbital data calculated via Simulink, avoiding excessive and unnecessary calculations related to attitude. Thus, as shown in Figure 14, the user selects the desired number of orbits, and the orbit is displayed on a world map after selecting the "Plot" button. An algorithm was also implemented to display the date and time of the body when it passes through a specific area, simulating the satellite's telecommunications antenna. In the example below, an area around the city of São Paulo was selected.

Furthermore, another tab related to the satellite's orbit was added, but disregarding disturbances, as occurred in the 2D plot in Figure 15, and with calculations performed via Matlab script. In this case, a 3D orbit will be plotted, showing the Earth's inertial coordinate system, and with greater visual appeal. The initial parameters required are: eccentricity, Right Ascension of Ascending Node, Perigee Argument, True Anomaly, and semi-major axis, all of which are Keplerian orbital parameters. Figure 16 depicts the final result of this plot.

CONCLUSIONS

It was possible to develop a graphical interface covering all the functionalities of the previously developed dynamic simulator, to validate the ADCS of the CASSTOR mission, in partnership with the Paris Observatory. The input data for simulation is already organized and validated, design mostly developed, and results presented to the user. The code is optimized, directories with organization logic according to the function of each of the files for the operation of the interface, rotational and orbital dynamics available

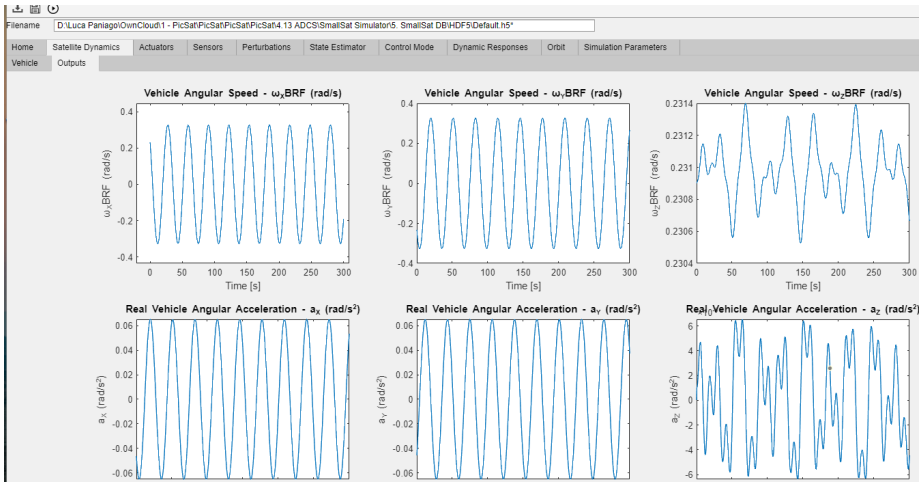

Figure 12. Plots of the velocity and angular acceleration of the simulated satellite in the "Satellite Dynamics" tab

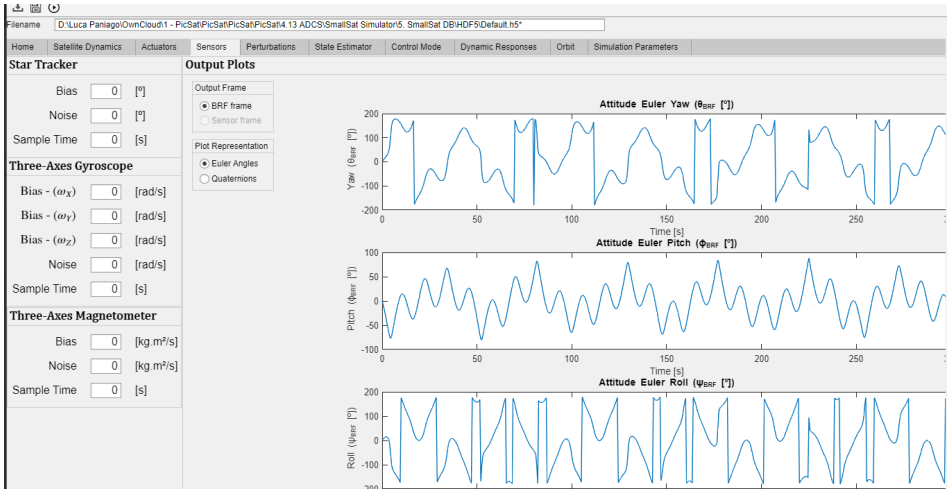

Figure 13. Plots of the attitude in Euler angles read by the sensors in the "Sensors" tab

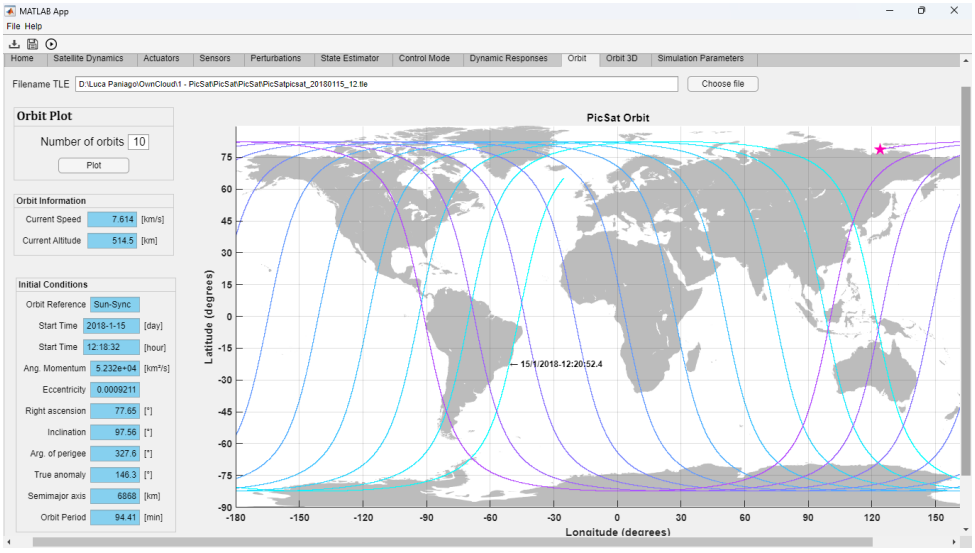

Figure 14. Detailed orbit plots

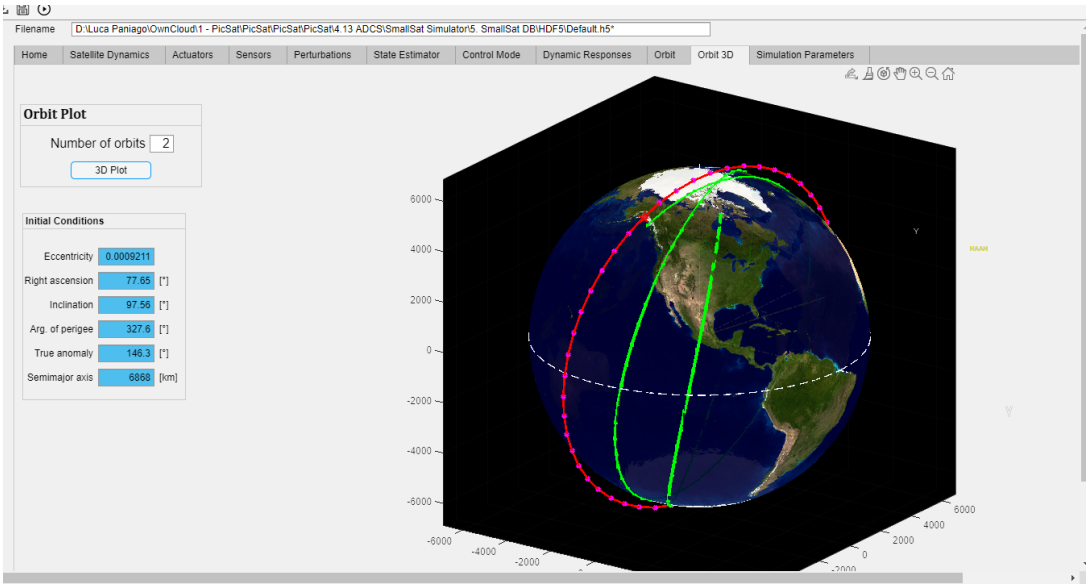

Figure 15. 3D orbit plot

for visualization, in addition to having new tools, such as estimating the satellite's inertia tensor, assisting the user in the development of their control project in the various stages of development.

However, there are still improvements to be made, such as the addition of the state estimator, which was not previously included in the dynamic simulator, the addition of more control modes, and making the user experience on the interface even more intuitive.

THANKS

The authors would like to thank CNPQ (National Council for Technological Development), process number 407381/2022, which finances the development of the simulator. They would also like to thank the project team members for their contribution: Lucas Wu Jiajun, Karoline Carvalho Bürguer and Cauê Garcia Menegaldo

REFERENCES

[1] Menegaldo, C. G. et al. PicSat's Enduring Legacy. Probing the Flight of a Small Astronomical Satellite. Publications of the Astronomical Society of the Pacific, v. 134, mar. 2022.

[2] Menegaldo, C. G. 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. Tese (Mestrado) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2020.

[3] Nowak, M. et al. Short life and abrupt death of PicSat, a small 3U CubeSat dreaming of exoplanet detection. Space Telescopes and Instrumentation 2018: Optical, Infrared, and Millimeter Wave, SPIE, v. 10698, jul. 2018. Disponível em: http://dx.doi.org/10.1117/12.2313242.