

IAC-22,C1,IP,18,x73772

TudatPy - a versatile open-source astrodynamics software package for space education

Kevin Cowan^{a*}, Filippo Oggioni^b, Geoffrey Garrett^b, Miguel Avillez^b, Joao Encarnacao^b, Marie Fayolle^b, Jonas Hener^b, Marc Naeije^b, Maarten van Nistelrooij^b, Ron Noomen^b, Michael Plumaris^b, Jeremie Gaffarel^b, Sean Cowan^b, Dominic Dirkx^b

^a *Space Department, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS, Delft, Netherlands* E-mail: k.j.cowan@tudelft.nl

^b *Space Department, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS, Delft, Netherlands*

* Corresponding author

Abstract

The TU Delft Astrodynamics Toolbox (Tudat), is an open-source astrodynamics toolbox written in C++ with a focus on numerical state propagation. It has been under development by staff and students of the Astrodynamics and Space Missions (AS) section of Faculty of Aerospace Engineering at Delft University of Technology (TUD) for more than ten years. During the process, Tudat has become a key part of the Space Exploration MSc curriculum at TUD. Moreover, Tudat has been used in dozens of MSc projects, 5 PhD projects, and has contributed to about 20 peer-reviewed publications.

Tudat provides both numerical state propagation and state estimation functionalities, organized in a modular setup. The design of Tudat ensures mutual consistency between the various environmental and dynamical models. Further development of Tudat is currently underway or planned in several categories, such as the integration with Machine Learning frameworks, the development of Precise Orbit Determination functionalities with real mission data, and the addition of more mission design and optimization techniques, including the integration with ESA's PaGMO/PyGMO toolbox.

Despite a continuous growth seen in the user base, the Tudat developer team has made the decision to create a Python interface for the software, motivated by the necessity of providing a more accessible and user-friendly tool that can be used not only by students and professionals, but also by space enthusiasts. Indeed, as a modern, high-level programming language, Python provides lower barriers to entry for users, significantly increasing the value of the Tudat kernel in education and academia. Named "TudatPy", this project has now become an ongoing effort to improve the accessibility of Tudat and to further develop the Application Programming Interface (API) for integration with external optimization and machine learning frameworks, while fully exploiting the Python language.

This paper thus demonstrates the use of the TudatPy API to 1) carry out an arbitrary optimization routine for a spacecraft mission using analytical methods, 2) perform the numerical simulation of the determined mission profile, 3) demonstrate a numerical analysis involving various integration schemes, acceleration models, and propagation methods, and 4) highlight various possibilities for uncertainty analysis and simulated trajectory estimation.

As an open-source project, TudatPy, and its installation and documentation, can be found at:

<https://tudat-space.readthedocs.io/en/latest/>

DO NOT EDIT THIS MAIN FILE. TO ADD YOUR MATERIAL, PLACE IT IN A SEPARATE .TEX FILE, WHICH WILL THEN BE IMPORTED USING AN \INPUT{} STATEMENT.

Nomenclature

[Input required here.]

Acronyms/Abbreviations

[Input required here.]

1. What is TudatPy?

[Input required here.]

2. What are its key features?

From the point of view of a user, what can you do with it? The primary user, for the purposes of this paper, is assumed to be: - Someone interested in using TudatPy in a way similar to that of a TU Delft Space instructor, so for education with a strong basis in research.

Features to mention:

Advantages / USPs to highlight:

- Modularity
- Fidelity
- Usability, particularly with the Python interface as well as how this allows the use of the vast range of Python tools now available (eg. Numpy, SciPy, Matplotlib, scikit-learn, ...)

To do: []As features are discussed, list published work (or educational settings) in which the various features have been applied.

[Input required here.]

3. Example applications

[Input required here.]

3.1 Analytical analyses

Proposal:

Two-part presentation:

1. Single arc propagation
2. MGA-DSM

[Input required here.]

3.2 State propagation

[Input required here.]

3.3 State estimation

[Input required here.]

3.4 Trajectory Optimization

[Input required here.]

- Without extensively advertizing for PaGMO [or/PyGMO], mention the integration with PaGMO and the key advantages of its integration, such as its open-source nature, successful track-record, ...

4. Get started with Tudat

A brief indication of where to find it, how to get started “playing” with it, how to contribute to development, ...