The Journal of Open Source Software

SimuPy Flight Vehicle Toolkit

Benjamin W. L. Margolis[1] **and Kenneth R. Lyons**[1]

1 NASA Ames Research Center, Systems Analysis Office

DOI: 10.21105/joss.04299

Software

- Review
- Repository
- Archive

Editor: Prashant K Jha

Reviewers:

- @athulpg007
- @aliaksei135

Submitted: 27 December 2021
Published: 07 July 2022

Summary

Vehicle flight simulation is an important part of the innovation of aerospace vehicle technology. Built on the SimuPy modeling and simulation framework (B. W. Margolis, 2017), the SimuPy Flight Vehicle Toolkit provides a modular framework for rapid implementation of simulations for novel flight vehicle concepts, such as hypersonic re-entry vehicles or urban air mobility vehicles. The open source repository of the source code includes implementations for the sixteen atmospheric test cases defined by the NASA Engineering Safety Center (NESC), which serve as validation of the simulation framework and examples of its usage.

Statement of Need

The NESC has identified and addressed the need to verify flight vehicle simulations through their work on the flight simulation test cases (Murri et al., 2015). In that work, the NESC established flight vehicle simulation test cases to compare and validate a suite of simulation tools, several from within NASA and one external, open-source tool. Implementations of the NESC test cases via the SimuPy Flight Vehicle Toolkit's API help verify its correctness and demonstrate its effectiveness in succinctly constructing flight vehicle simulations.

One author has used a precursor to this software package to simulate control system performance for a novel mechancially deployed hypersonic entry vehicle (D'Souza et al., 2019, 2021; B. Margolis et al., 2021; B. W. Margolis, Okolo, Nikaido, et al., 2019; B. W. Margolis, Okolo, & D'Souza, 2019; B. W. Margolis et al., 2020; Okolo et al., 2020).

Description

The SimuPy Flight Vehicle toolkit provides a modular programming interface for specifying vehicle and planet characteristics, such as aerodynamic coefficients and a gravity model, depicted in Figure 1. Equations of motion are composed in blocks using the SimuPy library (B. W. Margolis, 2017), an open source alternative to Simulink. These blocks can be formed into a standalone block diagram to simulate free behavior of the vehicle, or they can be incorporated into a more complex model. Implementations of the NESC test cases provided with the toolkit demonstrate usage for increasingly complex models, from a free-falling sphere in Earth atmosphere to a maneuvered F-16.

The Journal of Open Source Software

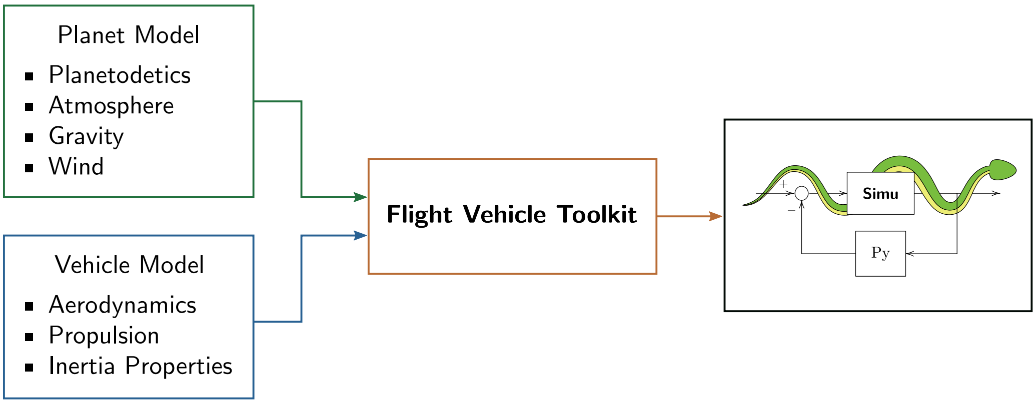

Figure 1: Simupy Flight Vehicle Toolkit architecture

The SimuPy Flight Vehicle Toolkit leverages open source scientific computing tools to implement an efficient simulation framework for flight vehicles in Python. Differential equations are solved using SciPy's wrappers for standard Fortran implementations (Jones et al., 2001--). Equations of motion for the inertial state of a rigid-body model of the vehicle representing the position, orientation, and their corresponding rates for integration are developed using the SymPy symbolic library (Meurer et al., 2017) and implemented using code generation. Kinematics equations are implemented through symbolic definition and code generation. Open-source scientific libraries are leveraged where possible, such as solving the inverse geodesy problem (Kerkwijk et al., 2020) and implementing a standard atmosphere model (Bell, 2016 -- 2021). The library also provides a parser for the American Institute of Aeronautics and Astronautics's (AIAA) simulation description mark-up language standard (Jackson & Hildreth, 2002) using code generation. Aerodynamic data table interpolation is implemented using ndsplines (B. W. Margolis & Lyons, 2019).

References

Bell, C. (2016 -- 2021). *Fluids: Fluid dynamics component of chemical engineering design library*. https://github.com/CalebBell/fluids

D'Souza, S. N., Alunni, A., Yount, B., Okolo, W., Margolis, B., Johnson, B. J., Hibbard, K., Barton, J., Hawke, V. M., Hays, Z. B., & others. (2021). Pterodactyl: System analysis of an asymmetric and symmetric deployable entry vehicle for precision targeting using flaps. *AIAA SciTech 2021 Forum*, 0762. https://doi.org/10.2514/6.2021-0762

D'Souza, S. N., Okolo, W., Nikaido, B., Yount, B., Tran, J., Margolis, B., Smith, B., Cassell, A., Johnson, B., Hibbard, K., & others. (2019). Developing an entry guidance and control design capability using flaps for the lifting Nano-ADEPT. *AIAA Aviation 2019 Forum*, 2901. https://doi.org/10.2514/6.2019-2901

Jackson, E. B., & Hildreth, B. (2002). Flight dynamic model exchange using XML. *AIAA Modeling and Simulation Technologies Conference and Exhibit*, 4482. https://doi.org/10.2514/6.2002-4482

Jones, E., Oliphant, T., Peterson, P., & others. (2001--). *SciPy: Open source scientific tools for Python*. http://www.scipy.org/

Kerkwijk, M. van, Tollerud, E., Woillez, J., Robitaille, T., Bray, E. M., Valentino, A., Sipőcz, B., Droettboom, M., Deil, C., Seifert, M., Conseil, S., Aldcroft, T., Price-Whelan, A., StuartLittlefair, Lim, P. L., Sulzbach, B., Beaumont, C., Cara, D., Crichton, D., … Šumak, J. (2020). *Liberfa/pyerfa v1.7.0* (Version v1.7.0) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.3940699

Margolis, B. W. (2017). SimuPy: A Python framework for modeling and simulating dynamical systems. *J. Open Source Software*, *2*(17), 396. https://doi.org/10.21105/joss.00396

Margolis, B. W., Ayoubi, M. A., & Joshi, S. S. (2020). Nonlinear model predictive control of reentry vehicles based on takagi-sugeno fuzzy models. *The Journal of the Astronautical Sciences*, *67*(1), 113–136. https://doi.org/10.1007/s40295-019-00191-2

Margolis, B. W., & Lyons, K. R. (2019). ndsplines: A python library for tensor-product b-splines of arbitrary dimension. *Journal of Open Source Software*, *4*(42), 1745. https://doi.org/10.21105/joss.01745

Margolis, B. W., Okolo, W. A., & D'Souza, S. N. (2019). Control design & sensitivity analysis for a deployable entry vehicle with aerodynamic control surfaces. *70th International Aeronautics Congress*.

Margolis, B. W., Okolo, W. A., Nikaido, B., Barton, J. D., & D'Souza, S. N. (2019). Control and simulation of a deployable entry vehicle with aerodynamic control surfaces. *AAS/AIAA Astrodynamics Specialist Conference, Portland, ME*.

Margolis, B., Okolo, W., D'Souza, S. N., & Johnson, B. J. (2021). Pterodactyl: Guidance and control of a symmetric deployable entry vehicle using an aerodynamic control system. *AIAA Scitech 2021 Forum*, 0764. https://doi.org/10.2514/6.2021-0764

Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., Rathnayake, T., Vig, S., Granger, B. E., Muller, R. P., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., … Scopatz, A. (2017). SymPy: Symbolic computing in python. *PeerJ Computer Science*, *3*, e103. https://doi.org/10.7717/peerj-cs.103

Murri, D. G., Jackson, E. B., & Shelton, R. O. (2015). *Check-cases for verification of 6-degree-of-freedom flight vehicle simulations* (TM-2015-218675). NASA.

Okolo, W. A., Margolis, B. W., D'Souza, S. N., & Barton, J. D. (2020). Pterodactyl: Development and comparison of control architectures for a mechanically deployed entry vehicle. *AIAA SciTech 2020 Forum*. https://doi.org/10.2514/6.2020-1012