Deutscher Luft- und Raumfahrtkongress 2024
DocumentID: 630453

Investigation of the dynamic behavior of two satellites during and after separation using the example of BIROS and BEESAT-4

F. Schero,* J. Hootz,† S. Maaß,‡ L. Gassama,§ Dr. H. Adirim¶

Abstract

This paper details the process and results of a study course at the Technische Universität Berlin that simulated the separation of two satellites using the example of BIROS and BEESAT-4 in order to investigate the resulting dynamic behavior. A simulation was developed that calculated the duration of the separation, the position of the satellites, the velocity during and after separation, the rotation of the spacecrafts and the associated impulses. Additionally, it is capable of propagating the orbits of the two satellites after the separation and of calculating the distance between them. The results were validated using the commercial software FreeFlyer. This showed successful simulation of the dynamic behavior of two satellites after separation, with a focus on the effects of the spring-loaded separation containers used during the mission on the flight characteristics of both satellites.

I Introduction

The Technische Universität Berlin developed an 1U-CubeSat named BEESAT-4 that was launched in cooperation with the German Aerospace Center (DLR). The idea was to eject the CubeSat from the BIROS satellite with a separation container, where the satellite would be ejected with a rail-guided spring system. This took place on September 9th, 2016 at 1:00 pm Central European time. BEESAT-4 was ejected at a height of 515 km above the Norwegian island group Spitzbergen. [1]

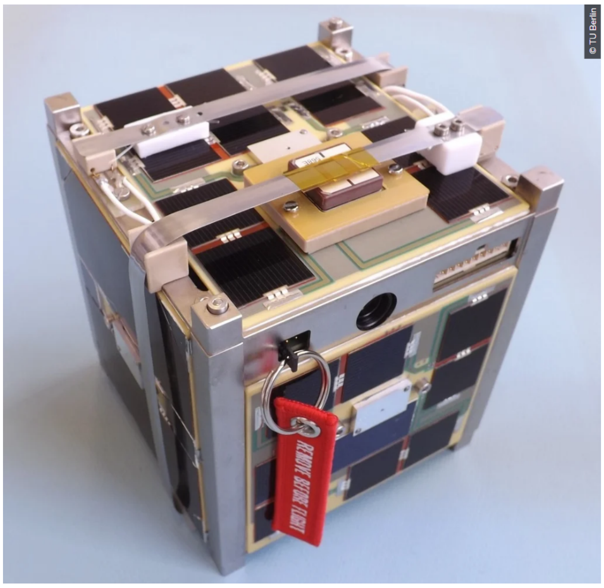

Figure 1: BEESAT-4 before launch [2]

The BEESAT-4 satellite itself, which is shown in figure 1, was placed into the separation container of BIROS by hand, which can be seen in figure 2. The purpose of this mission was to test the environment and behavior of two satellites launching together and then separating in space. Since there had not been many missions similar to this one before, the impact of the separation on the satellites could not be easily estimated. Due to the compact size of the CubeSat, the separation container could be placed within the BIROS satellite. The mission objective was to eject BEESAT-4 from BIROS after the latter reached its designated orbit. This approach presented several benefits. Firstly, BEESAT-4 did not require a dedicated slot on a launch vehicle for launch into space. Secondly, valuable data could be collected on the dynamic behavior of both satellites during and after separation, enabling research into this phenomenon. Because the separation container, in which BEESAT-4 was placed, was not located at the center of mass of BIROS, a rotation effect was expected. Additionally, due to the impulse of the ejection, slight changes in the orbit of BIROS and the resulting orbit of BEESAT-4 were expected. Since the exact impact of ejecting BEESAT-4 from BIROS was uncertain, and it could not be predicted how much BIROS would rotate and how significant the impact on its orbit would be, a key goal was to investigate the behavior of the satellites as the ejection energy acted as a force vector upon them.

Figure 2: BEESAT-4 being placed into BIROS' separation container [1]

doi: 10.25967/630453

The focus of this modeling approach was to simulate this separation process and investigate the dynamic behavior of the satellites after the separation. Data on separation speed, rotational speed, acceleration, position, momentum, and more were collected. Additionally, propagation of the satellites orbits after separation was performed to estimate the effect of the ejection on the orbits. Although the simulation can show the dynamic behavior of many missions, the focus was placed on the BIROS and BEESAT-4 mission from the DLR and TU Berlin. The main mission parameters were simulated using a Python script and then validated with regard to values such as Kepler elements, altitude, distance, and speed. For validation, the commercial software FreeFlyer [3] was used, in which the separation of the two satellites was simulated in parallel. The same parameters were transferred to the Python code, and the results were compared with those from FreeFlyer for validation. Using inputs that describe the positions of the spacecraft in orbit relative to each other, it was possible to simulate the mission and the separation of the two satellites and calculate important parameters such as resulting rotation, velocity changes, impulses, etc. By specifying the two-line elements (TLEs) of the initial orbit of the mother ship — in this case, the BIROS satellite — the effect on the orbit itself can be calculated and propagated over any period of time. The focus was to simulate a separation of two spacecraft to a satisfactory extent. The simulation delivers accurate results with more or less comparable accuracy to the commercial solutions. In the future, this could help TU Berlin to better predict orbital separation processes and optimize future missions. In the future, ride-share missions and the deployment of large constellations of nano- and cubesats from a single launcher will likely increase in relevance and strategic importance to the space industry [4]. As such, robust simulations of the effects of spring-loaded separation systems on orbital parameters and behaviour of individual satellites will gain importance. The simulation aims to provide a starting point for others to investigate these relationships while providing a very low barrier to entry.

II Assumptions

Several assumptions were made to simplify the simulation and to focus on the impact of the separation event on the orbital mechanics of the satellite, rather than simulating the separation mechanism itself. One of the primary assumptions was the simplification of the satellites geometry. The satellites were modeled with a homogeneous mass distribution, meaning the mass was assumed to be evenly distributed throughout the structure. This simplification made it easier to calculate key properties, such as the center of mass, which is used to determine the satellites behavior during and after separation. Additionally, the satellites were modeled as cuboids, and attachment parts such as antennas were not considered. The exact dimensions of the satellites were taken into account, and the position of the separation container—and therefore the force vector of the separation—was evaluated using the accommodation model of BIROS.

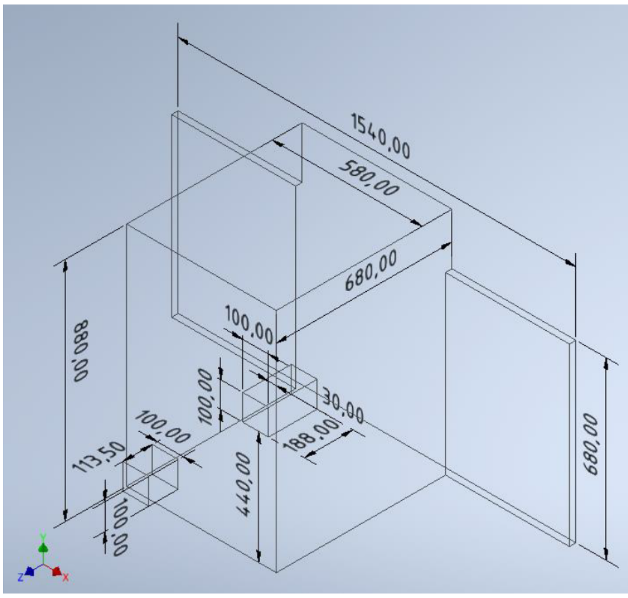

Figure 3: Dimensions of BIROS and BEESAT-4 in mm

The measured dimensions were then transferred into a CAD-Model of the satellite, to evaluate the exact position of the force vector coming from the separation container. The CAD-Model with it's dimensions in mm is shown in figure 3.

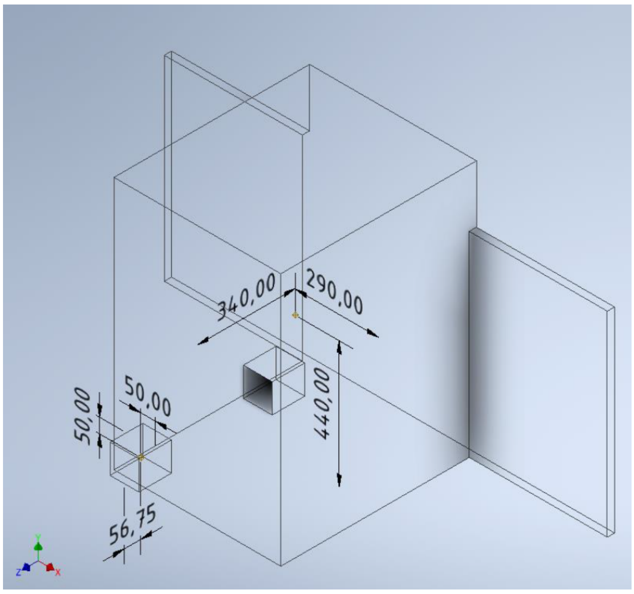

Figure 4: Center of mass of BIROS and BEESAT-4 measured in mm

Additionally, the CAD-Model was used to evaluate the center of mass of BIROS and BEESAT-4, which is shown in figure 4.

The impact of these simplifications is considered to be low in regards to the overall analysis of the effects on flight characteristics when already in orbit. The separation container, which is a rail system with a spring, was also simplified during the modeling process. The internal friction of this rail system was not taken into account, as the focus was on the effect of the separation on the orbit rather than on simulating the separation mechanism. Additionally, a linear spring extension was assumed. Each of these simplifications were made to reduce the complexity of non-essential factors. The key focus was to understand the satellites resulting orbits after separation.

III Implementation

The simulation was realized with a Python script in the JupyterLab IDE [5]. Several open-source libraries were used for respective parts of the script. The open-source approach allows for easy adaptation and opens up the possibility of further development. Because of requirements of the libraries used, the Python version is limited to between 3.7 and 3.10. The simulation relied heavily on open-source libraries created and supported by other users. The most important ones for this work were poliastro [6], astropy [7] and skyfield [8]. The script is divided into several sections, each representing an individual part of the required calculations. The sections can be labelled as: 1) inertia calculations, 2) separation event and 3) orbital parameters and propagation. A high-level flowchart of the script is shown in figure 5.

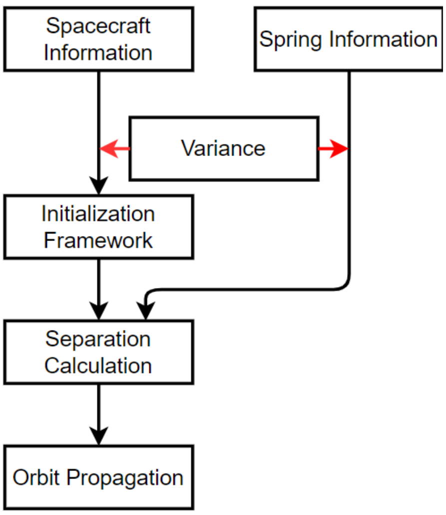

Figure 5: High-level flowchart of the script

As has been described, this modeling effort focused on the BIROS and BEESAT-4 separation event. As for information required by the system in order to run the simulation successfully, it is for the most part concerned with defining the physical characteristics of the spacecraft. Thus, information on the mass and its distribution for both spacecrafts is vital. Geometric information on the spacecraft is also required. These two points result in a requirement for the user to specify both the dimensions of the spacecraft as well as the spacial relations between their respective center of mass and the axis along which the spring force will act. As for the separation container, the user is required to input information about its performance characteristics such as (nominal) ejection force, the length of the relaxed spring as well as its compression distance. In addition, the orbital parameters are required. These are to be integrated into the code in the form of Two-Line-Elements (TLE), following the NORAD TLE format. Using this information, the script calculates a center-of-mass for each satellite body, which in turn are represented by cuboids with adjustable side lengths. The mass of BIROS was estimated as 130 kg, with BEESAT-4 weighing in at approximately 1kg. The separation was carried out by a spring-loaded canister separation system. Information was gathered from the responsible engineers directly during a visit to the premises of Astro- und Feinwerktechnik Adlershof GmbH, where the separation container was designed and manufactured, as well as the DLR Institute of Optical Sensor Systems in Berlin where the BIROS satellite was developed and integrated.

Figure 6: Astro- und Feinwerktechnik Adlershof PSL1U and PSL3U Cubesat Launchers [9]

The separation impulse is provided only by the spring, which is set before launch of the spacecraft. In order to calculate the potential energy stored in the spring, its spring constant is first calculated as an averaged value:

$$k_{avg} = \frac{\frac{1}{2} * (F_{initial} - F_{end})}{l_{comp}} = 79.24 \frac{N}{m} \tag{1}$$

Where k_{avg} is the averaged spring constant, $F_{initial}$ is the force initially provided by the spring, F_{end} is the force at the final spring position and l_{comp} is the total length of compression. The ejection forces used in the simulation can be found in table 1. Using the aforementioned mass and inertia calculations for both spacecraft, and defining a compression length for the spring, these results allow for determination of separation velocities and resulting spacecraft tumbling.

For every time step during the separation event, the script will calculate the velocity of both spacecraft according to the following formulation:

(2) $$v_2 = v_1 + a_1 * \Delta t$$

Where v_2 and v_1 are the velocities of the spacecraft in the current and the next time step, a_1 is the acceleration and Δt is the time step. The above formulation assumes that the axis of separation is in line with both spacecraft centers of mass. In reality, this is not the case. In order to find the distribution of the potential energy provided by the spring between translational and rotational kinetic energy, the user is able to set the force vector. Depending on the distance of the axis along with the force is applied to the axis along which the CoG lies, different fractions of potential spring energy are distributed to the different forms of kinetic energy present in the separated system.

(3) $$a_1 = (F_1/m_1) * (1 - (d_1/100))$$

The above formulation relates the acceleration of the spacecraft to the relative position of the separation force vector and the center of mass of the vehicle: d_1 is distance between those points normalized over the length of the satellite body, m_1 is the vehicle mass, F_1 is the spring force and a_1 is the resulting acceleration. During the separation event, both spacecraft are subject to constant acceleration, which assumes linear spring forces over the expansion distance. In the simulation, this constant acceleration is present until the script detects that the distance between the two spacecraft is greater than their initial separation and the spring expansion distance combined. Once this distance is reached, the separation event is considered complete. The orbital propagation portion of the script utilized the aforementioned libraries. After generating an orbit from the provided TLE for the host spacecraft, data about the velocity and direction of the spacecraft are exported from the orbit instance. The results of the separation simulation are then added onto the obtained values for velocities and afterwards the now updated data is used to again generate an orbit using the skyfield library. These newly generated orbital parameters are finally used to propagate the orbit of the now independently flying spacecraft. This propagation is open-ended in the time domain. Of course, the longer the simulation period, the larger any present errors might grow due to numerical inaccuracies and floating-point issues present in the simulation.

IV Limitation

As the dynamics of the separation of two satellites were to be simplified, consequently input parameters influence the accuracy of the results.
Another challenge was calculating the correct spring constant. The separation container in which the BEESAT-4 was located and from which it was released consists of a rail-guided spring system. To improve the determination of the spring constant, the friction of the rail system was neglected. Additionally, a sensitivity analysis was done in order to better evaluate the impact of the spring on the ejection energy. Fluctuations of $\pm 20\%$ of the nominal value of $79,24\frac{N}{m}$ of the potential energy of the system were implemented in order to estimate the effect of the strength of the spring on the behavior of the satellites.

Sim 1	63.39 N/m	Sim 6	83.20 N/m
Sim 2	67.35 N/m	Sim 7	87.16 N/m
Sim 3	71.32 N/m	Sim 8	91.13 N/m
Sim 4	75.28 N/m	Sim 9	95.09 N/m
Sim 5	79.24 N/m		

Table 1: Used spring constants taken sensitivity analysis into account

The simulation was executed with the resulting spring constants shown in table 1.

	12 hours	**24 hours**
-20%	15.16 km	35.42 km
-10%	16.04 km	37.47 km
Reference Spring	16.93 km	39.55 km
10%	17.76 km	41.48 km
20%	18.57 km	43.38 km

Table 2: Development of the distance between the two satellites for varying spring constants after 12 and 24 hours

The distance between the satellites, which can be seen in table 2 shows that major effects on the separation and the subsequent behavior of the satellites must be assumed. The simulation showed a difference in distance between the two satellites of approximately 8 km between the minimum and maximum ejection energies after 24 hours in orbit. Therefore, the ejection energy of the separation container is a crucial part of the process, which has to be kept in mind.
The separation simulation written in Python can therefore also be beneficial for initial research of the effect of varying ejection energies. It can provide quick and close initial results with regard to input variables, and thus possible sources of error can be calculated even before the launch of such a mission.
Additionally, the simulation is capable of calculating other data such as rotation rates.
The propagator then uses the entered two-line elements of the initial orbit and generates an orbit element from these parameters. Then, the position changes caused by the separation and atmospheric influences are added to the initial orbit and a new orbit element with the associated Kepler-Elements will be created. This can be done for a desired amount of time.

V Results

Once the spring of the separation container begins expanding, the simulation is starting to calculate the results, which will be presented in this chapter.

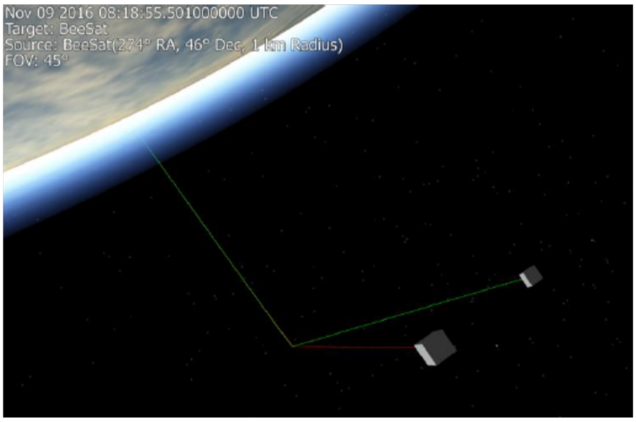

Figure 7: 3D-Visualisation of the separation in FreeFlyer

The ejection of the BEESAT-4 satellite out of BIROS will cause the satellites to move away from each other, which is represented by the 3D-Visualisation in FreeFlyer in figure 7 and therefore the orbit parameters will change.

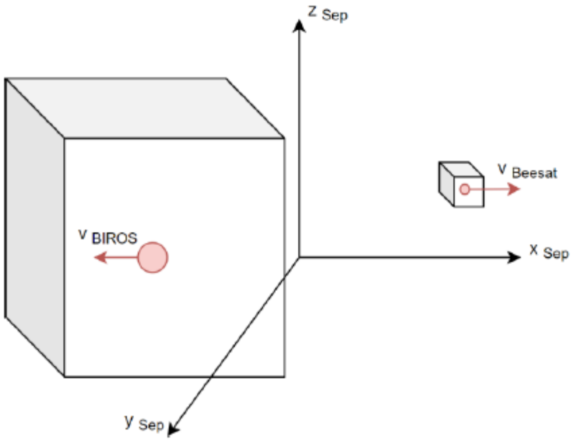

Figure 8: Position of the satellites in the local coordinate system right after separation

The positional changes of the satellites are initially calculated in a local coordinate system, which is shown in figure 8. These positional changes have to be merged with the calculated rotational changes and the existing orbital parameters. The values must be calculated and validated before giving an evaluation on the effect of the separation on the orbital parameters of the satellites.

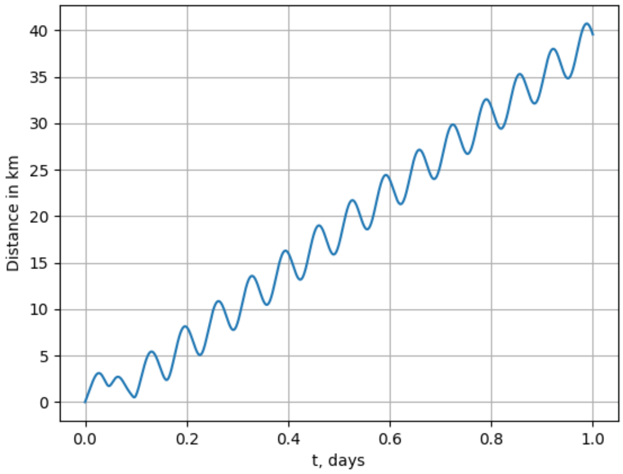

Figure 9: Distance of satellites over time

As shown in figure 9, the separation of both masses is described by a ascending sinus graph. The sinus curve is attributed to the elliptic shape of the orbit so the satellites have phases where they get closer to each other and phases where they increase their distance.

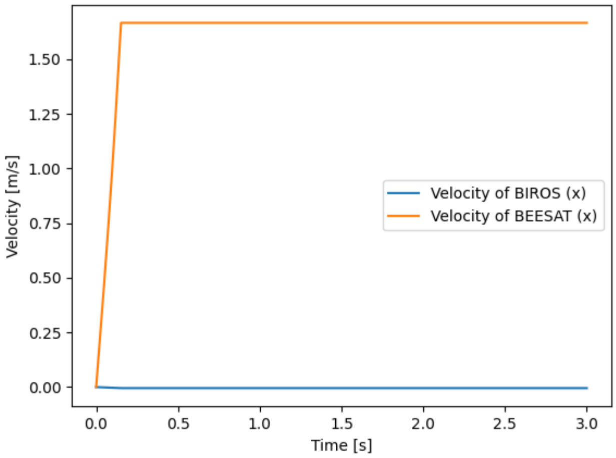

Figure 10: Velocities of the satellites over time

In figure 10 the velocity of both masses is shown. The spring is compressed at the start of the process and accelerates both spacecrafts as it expands. After approximately 0.2 seconds the separation takes place, which results in no further acceleration and gives both spacecrafts a constant velocity. The velocity change of BEESAT after separation amounts to $1.67m/s$ and the one of BIROS to $0.005m/s$. It can be seen that the velocity of BEESAT gets much higher than the one of BIROS, which can be explained with the lower mass of BEESAT.

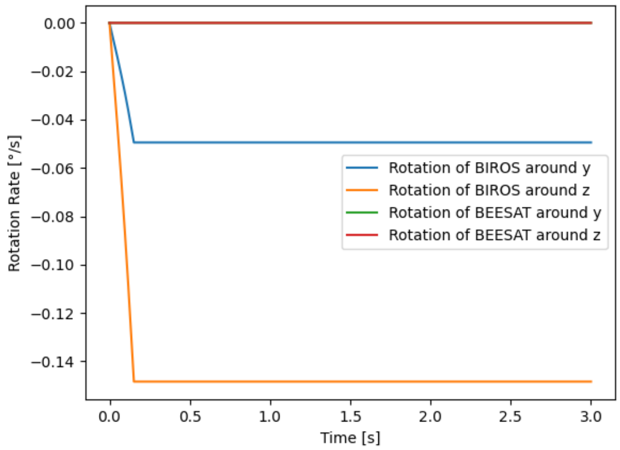

Figure 11: Rotation rate of the satellites over time

The rotation of the satellites is shown in figure 11. Ideally, the rotation rate of BEESAT is zero, because the force vector of the separation container is in line with the center of mass. The rotation rate of BIROS is relatively small, which can be justified by the high inertia of BIROS due to its mass. The satellite is experiencing rotational acceleration because the separation container is not in line with the center of mass. After the separation the acceleration is reduced to 0 for both spacecrafts. Since no effects of friction or other disturbances are taken into account, the rotational speed remains at constant value.

Since the calculation for the pulse uses mass and velocity the course of the graph is identical to the graphs of velocity. For the propagation of the orbit around earth the orbit propagator by PoliAstro has been used. Firstly, the Two-Line-Elements must be known in order to gather data about the position and velocity of the BIROS satellite. After calculating those vectors with the data changes by the simulation, there are now two pairs of vectors as input data. One combination of position and speed for the mothership and one pair of vectors for the separated satellite.

The simulation can display the orbits of the satellites. Due to the separation of the much lighter satellite, the orbit of BEESAT turned to more of an elliptic orbit.

p	Semi-Latus-Rectum
e	Eccentricity
i	Inclination
Ω	Longitude of the ascending node

Table 3: Kepler-Elements

To evaluate the impact of the separation on the orbit of the mothership, in this regard BIROS, the keplerian elements, who are described in table 3 are used. The results of the orbit of BIROS without- and with separation after 24 hours can be seen in table 4. It is noticeable, that the ejection of BEESAT-4, which was done in flight direction, has an impact on the semi-latus-rectum and the eccentricity of BIROS' orbit. Even with the little change in velocity of $0.005 m/s$ the impact on BIROS' orbit is noticeable and has to be kept in mind for the mission constraints.

BIROS	Without Separation	With Separation
p[km]	6892.31	6891.57
e	1.53e-02	1.12e-02
i[°]	97.43	97.43
Ω[°]	313.93	313.93

Table 4: Orbit parameters after 24 hours

The final results are the orbital parameters of both satellites after the separation and the propagation of their orbits. The propagation used data calculated during the 0.4 seconds of separation and combined it with the given data from the BIROS satellite to get the final vectors after the ejection. Data regarding position and velocity of both satellites were then calculated for each time step until the simulation ends. These values were converted into Kepler-Elements for further validation. For this process various parameters needed to be taken into account. Those parameters were the alternating speed and position of the satellites and the position of the spring mechanism, since it was not placed in the center of mass.

VI Validation

To begin the software validation process, an initial quality assessment was conducted to demonstrate its capability to be approximately comparable with commercial software in terms of accuracy. For that measure the software FreeFlyer has been used and a similar simulation was created. Since FreeFlyer does not contain an inbuilt method for displaying a satellite separation, the maneuver was simulated through a short pulse with the values of the earlier calculated speed and duration of spring expansion. As a parameter for the validation the Kepler elements were chosen. The difference between those parameters would determine how well the software is performing. It should be kept in mind that the percentage deviation may be high, even though the absolute value does not differ much. The reason for this lies in the extremely low value of those parameters for example eccentricity.

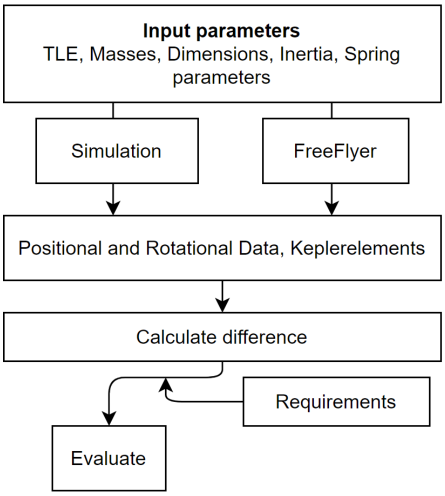

Figure 12: High-level overview of validation plan for the simulation

For the validation, a flow chart was created, which is shown in figure 12. When the TLEs, masses, dimensions, moments of inertia and spring parameters are available as inputs, these inputs get implemented in the simulation and FreeFlyer. Then, position vectors, rotational rates and Kepler Elements are calculated and the deviation between the simulation and FreeFlyer will be evaluated. When that is done, an final evaluation can be formulated.
To compare the simulation with the results from the commercial software FreeFlyer, a comparison in regards of Keplerian Elements were chosen. These Keplerian Elements were evaluated for BEESAT and BIROS in table 5 and 6.

	Simulation	FreeFlyer	Difference
p[km]	6891.26	6867.10	0.35%
e	1.13e-02	1.14e-03	0.89%
i [°]	97.42	97.50	0.08%
Ω [°]	313.93	315.15	0.39%

Table 5: Comparison of orbit parameters of BEESAT after 24 h

	Simulation	FreeFlyer	Difference
p[km]	6891.57	6857.44	0.49%
e	1.12e-02	1.13e-02	0.88%
i [°]	97.43	97.51	0.08%
Ω [°]	313.93	315.15	0.39%

Table 6: Comparison of orbit parameters of BIROS after 24 h

As seen in the tables, all the values are under 1% difference. Looking at the differences over time its noticeable that those difference take two possibilities of a course. Either the difference grows continuously or the difference follows a periodic pattern. Due to the low difference, the values can be considered as validated.

VII Conclusion and Outlook

This modelling effort, which was carried out over the course of two semesters at the TU Berlin, focused on the dynamics of two satellites during their in-orbit separation process. The main interest was to visualize such a separation and to understand how the satellites would behave. Initially, the simulation was intended to be as general in nature as possible, however during the development a more focused approach was chosen, better suited and adapted to capture the details of the BIROS / BEESAT-4 example. This mission was particularly suitable as the BEESAT-4 was a satellite manufactured by TU Berlin and was deployed from the BIROS satellite in cooperation with DLR. Nevertheless, the underlying code-base for calculating the dynamics and orbit propagation was kept general and can be applied to any other example in orbit. In order to have an additional basis for validation, the program FreeFlyer was used, which enabled the team to carry out a comparative evaluation of the results. Within the validation campaign, the question of how closely the script and commercial software overlapped in regards to their results was assigned greatest importance.Finally, accurate results were achieved. In conclusion, software has been developed as part of a student project that can visualize the dynamics of two satellites during a separation process. This software was applied to the example simulation of the mission already carried out by the DLR and TU Berlin and the associated BIROS and BEESAT-4 satellites. In the future, this simulation model can be extended to provide more functionality and improved performance. For example, the computing speed can be improved by adapting the source code. Other parameters, such as an in-homogeneous mass distribution or an implementation of an attitude control system, could also improve the simulation. As the BIROS AOCS was active during the separation during real mission, it would be beneficial regarding the accuracy to implement them at a later stage.

We want to thank those who helped us during the development of this work and its publication, especially our colleagues from the TU Berlin, the DLR Institute of Optical Sensor Systems, Astro- und Feinwerktechnik Adlershof GmbH and the team behind FreeFlyer.

References

[1] Deutsches Zentrum für Luft- und Raumfahrt (DLR). DLR-Feuersatellit BIROS entlässt Kleinsatellit BEESAT-4 erfolgreich ins All. https://www.dlr.de/de/aktuelles/nachrichten/2016/20160905_dlr-feuersate

[*] TU Berlin, finn.schero@campus.tu-berlin.de
[†] TU Berlin, j.hootz@campus.tu-berlin.de
[‡] TU Berlin, sebastian.christopher.maass@campus.tu-berlin.de
[§] TU Berlin
[¶] AI: Aerospace Innovation GmbH, harry.adirim@aerospace-innovation.com

llit-biros-entlaesst-kleinsatellit-beesat-4-erfolgreich-ins-all_19203, 2016. [Online; accessed 03-Sep-2024].

[2] TU Berlin. BEESAT-4. https://www.tu.berlin/raumfahrttechnik/forschung/aktuelle-projekte/beesat-4. [Online; accessed 03-Sep-2024].

[3] ai-solutions. FreeFlyer - Astrodynamics Software. https://ai-solutions.com/freeflyer-astrodynamic-software/, 2024.

[4] Michael Swartwout. Cubesats/Smallsats/Nanosats/Picosats/Rideshare(sats) in 2022: Making Sense of the Numbers. 2022 IEEE Aerospace Conference Proceedings, 2022.

[5] Jupyter. Jupyter. https://jupyter.org/, 2024.

[6] Juan Luis Cano Rodríguez et al. poliastro - Astrodynamics in Python. https://docs.poliastro.space/en/stable/, 2024. [Online; accessed 19-Sep-2024].

[7] The Astropy Collaboration et. al. The Astropy Project: Sustaining and Growing a Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core Package. https://www.astropy.org/, 2024. [Online; accessed 19-Sep-2024].

[8] Brandon Rhodes. Skyfield: High precision research-grade positions for planets and Earth satellites generator. https://rhodesmill.org/skyfield/, 2024. [Online; accessed 19-Sep-2024].

[9] Astro- und Feinwerktechnik Adlershof GmbH. PicoSatellite Launcher (PSL), Technical Specifications. Technical report, Astro- und Feinwerktechnik Adlershof GmbH, August 2016.