Professorship of Human Spaceflight Technology
TUM School of Engineering and Design
Technical University of Munich

Development of a Flight Data Evaluation Tool

Analyze and Evaluate Soyuz Simulator Docking Maneuver Metrics

Anton Woy, Supervisor: Prof. Dr.-Ing. Gisela Detrell

Introduction

The new Soyuz spacecraft simulators at the Professorship of Human Spaceflight Technology (based on University of Stuttgart heritage) will be used for research and from students in a dedicated Aerospace Masters Seminar in various flight scenarios. In order to evaluate and analyze the raw flight data provided by the simulator, a specialized software is needed.

Previously, a C++ tool[1] developed at the University of Stuttgart was used to evaluate docking maneuvers, mainly for research purposes. To address shortcomings caused by the different intended use of the old software, it was decided to redevelop the Flight Data Evaluation Tool with an increased focus on easy maintainability and operation in a university context. All relevant functionalities of the old software should be available, to allow a seamless transition.

Methods

The new tool is written in *Python*, which offers a wide range of scientific libraries and is easy to learn to allow future development by students, and is distributed under the MIT license.

The algorithm to detect flight errors has to treat translational and rotational control differently. Rotational velocities are automatically corrected by the Soyuz systems after a steering input. For translational maneuvers, the algorithm should not detect breaking maneuvers as an error. Also the translation on the axis towards the station (x-axis) has to be considered separately, since its ideal values for position and velocity are the only ones that are not zero-centered.

The following errors are considered by the algorithm developed for docking maneuvers:

Translation x-Axis	Translation y/z-Axis	Rotation x/y/z-Axis
Acceleration above *Ideal Approach Velocity* towards the station	Leaving zero positional offset with maneuver	
Further Acceleration despite being already above *Ideal Approach Velocity* towards the station	Increasing positional offset with maneuver in *positive* direction; exclude breaking maneuvers; initial offset NOT zero	Increasing positional offset with maneuver in *positive* direction; initial offset NOT zero
Acceleration away from the station	Increasing positional offset with maneuver in *negative* direction; exclude breaking maneuvers; initial offset NOT zero	Increasing positional offset with maneuver in *negative* direction; initial offset NOT zero

Table 1: Overview over considered errors for all Body Axes

A docking maneuver consists out of several phases with different characteristics (see Figure 1). To evaluate the flight, the software must detect the transition between the phases, with *Alignment* and *Approach* phase being the most challenging to differentiate. Stricter conditions are able to detect this phase transition in non-nominal flight scenarios, but are worse at generalizing. Therefore, the conditions are as tolerant as possible, analog to the old tool[1]:

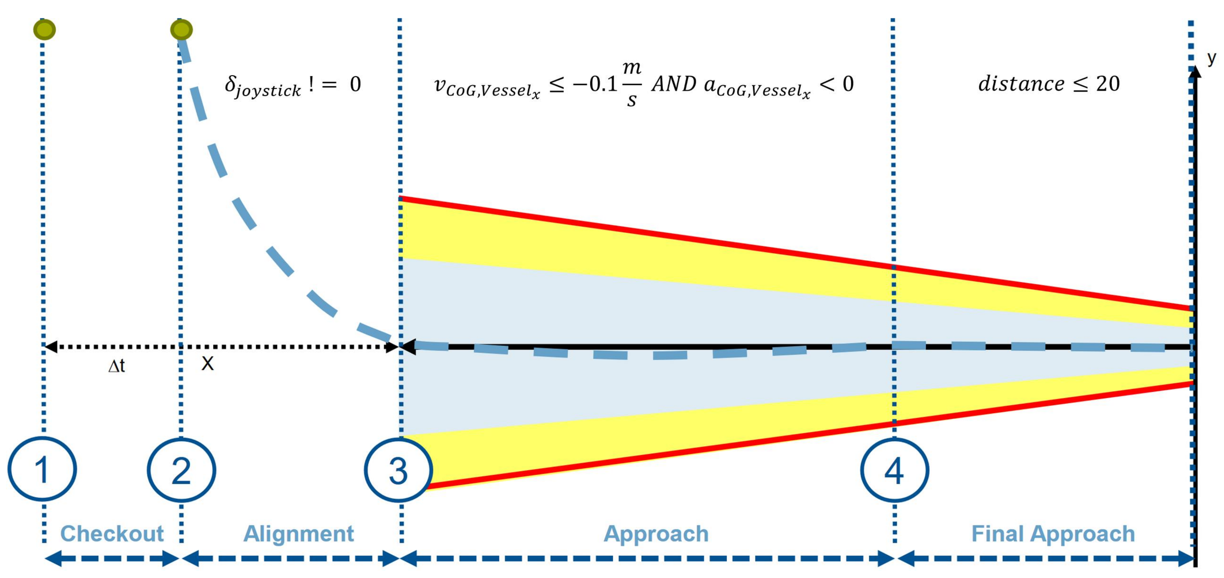

Figure 1: Overview Docking Maneuver Phases[2]

If the conditions for one or more phases are not met, fallback values are calculated so that they are evenly spaced between the phase transitions that could be calculated and the user is informed that a manual phase adjustment is required. Even if no fallback values had to be calculated, a manual input may be necessary, since the conditions for the different phases apply to its first appearance, which in some cases may not be the real phase transition.

Results

The new flight analysis Software Features the following Visualizations:

i. Translational offset Vessel to docking Port
ii. Translational velocity of Vessel Center of Gravity (CoG)
iii. Angular position of the vessel
iv. Rotation velocities of Vessel around Center of Gravity (CoG)
v. Translational controller inputs
vi. Rotational controller inputs
vii. Fuel mass in tank during maneuver
viii. Visibility of docking Port (angle between x-Axis of port and vessel periscope axis)
ix. Heatmaps of the translational movement in the y/z-plane for every flight phase

The plots i-vii were already part of the old software, viii and ix were newly developed.

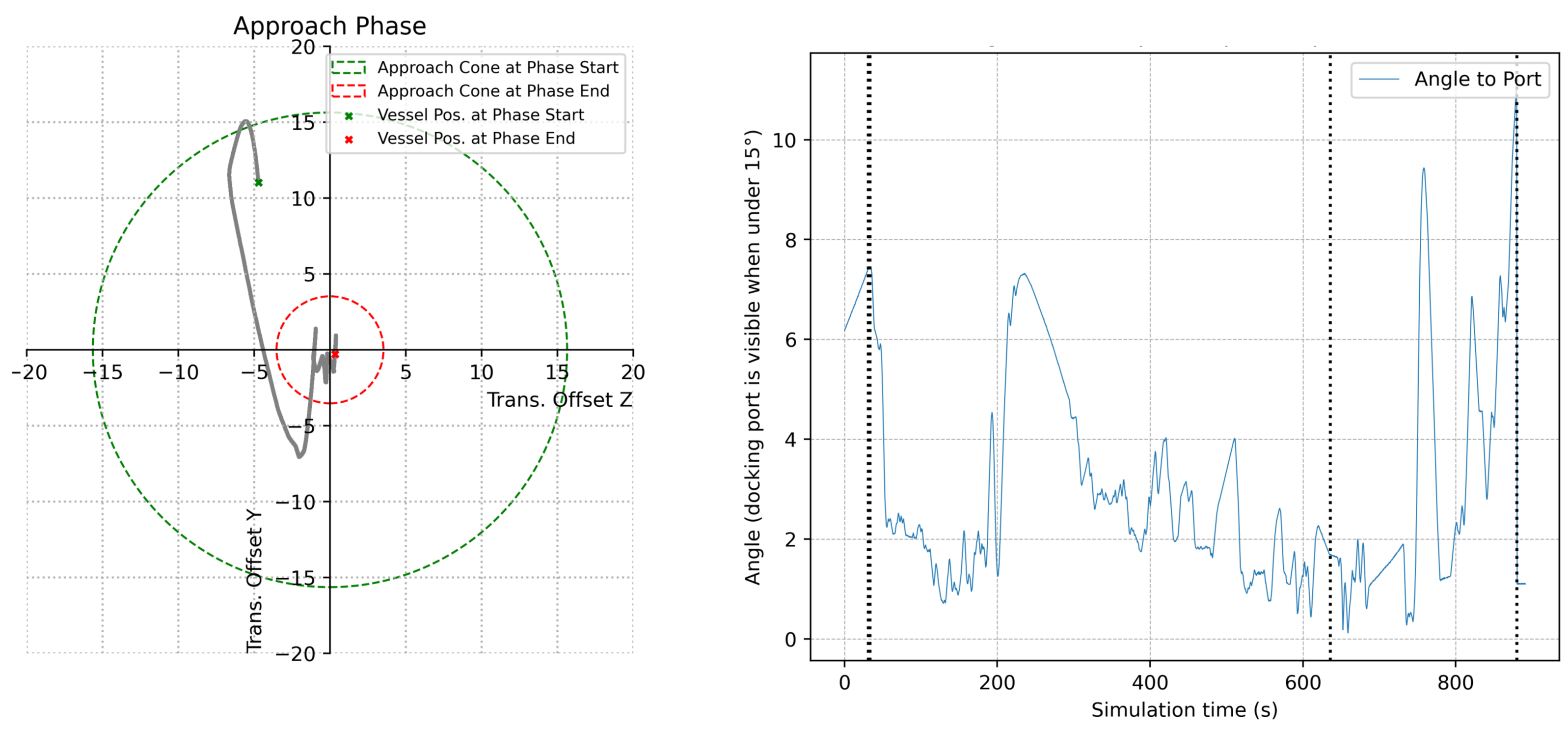

Figure 2: a.) Translational movement of vessel in y/z-plane during Approach Phase b.) Angle between x-Axis of port and vessel periscope axis

Figure 2a shows the lateral position of the vessel at the start and the end of the approach phase and the flight path between these two points. This visualization is available for each phase of the flight, as well as for the entire flight, and is extremely helpful in gaining a better understanding of the lateral offset of the vessel during a docking maneuver. The second new plot is featured in Figure 2b and shows the angle between the periscope axis and x-axis of the docking port. The port is visible as long as this angle is less than 15° and therefore inside the 30° field of view of the Soyuz periscope[3].

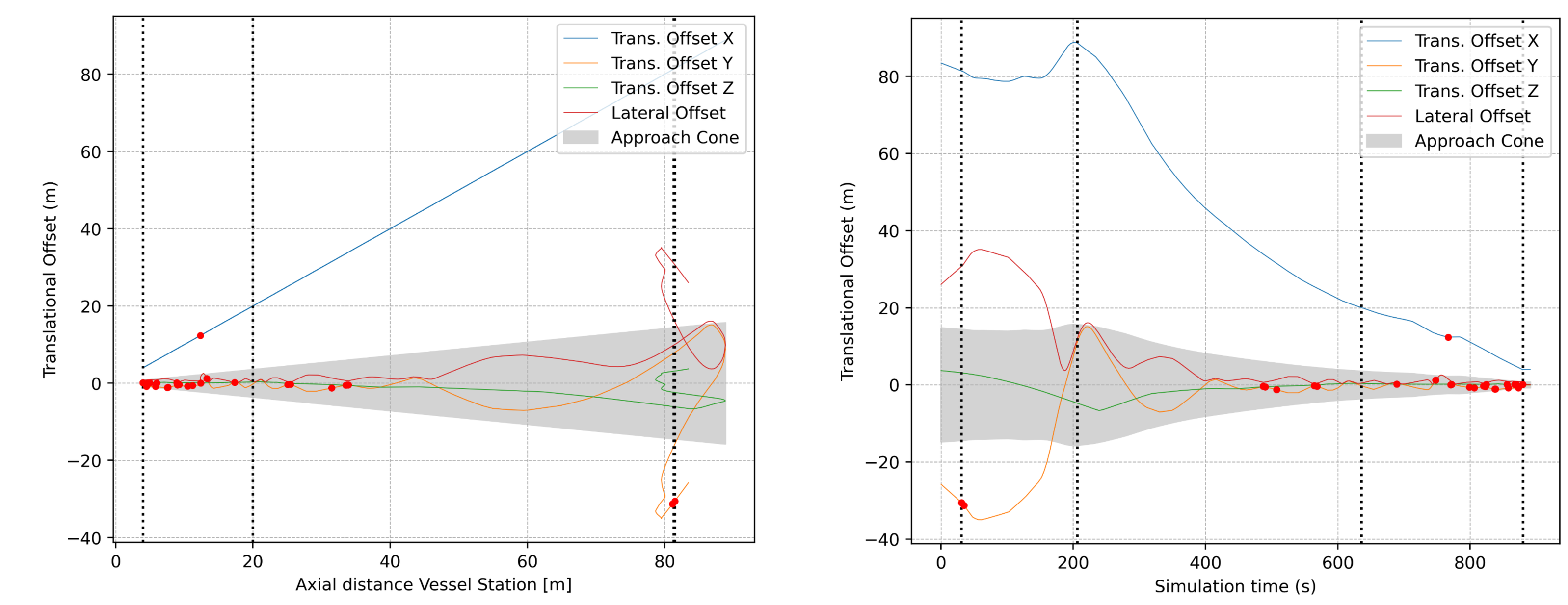

Figure 3: a.) Translational offset Vessel to docking Port over axial distance Vessel-Station b.) Translational offset Vessel to docking Port over Simulation Time

The new tool is now capable to display the plots i-viii not only over *simulation time* (Figure 3b) but also over the *axial distance* to the station (Figure 3a). The new visualisation better displays the approach cone and provides new insights in the data, but does not show the *Alignment* phase well, since ideally no axial movement should happen.

Discussion

The majority of the numerical flight metrics to asses pilot performance are either identically calculated by both tools or only differ slightly, see Table 2. The smaller deviations result from the different reference point in phase detection (Vessel-Port vs. Vessel-CoG). The evaluation values related to the detected flight errors account for the number of values with a percentage difference of more then 25%, since the two algorithms differ significantly. However, Table 2 proves that also the new Software can also be used for flight performance evaluation.

Percentage Difference	Identical	(0%, 5%]	(5%, 10%]	(10%, 15%]	(15%, 20%]	(20%, 25%]	(25%, ∞)
No. of Values	1609	1403	472	226	155	111	1314
Distribution	30.42 %	26.52 %	8.92 %	4.27 %	2.93 %	2.10 %	24.84 %

Table 2: Comparison of calculated evaluation values from 23 Check Flights to asses flight performance of both tools

For more complex flight trajectories, manual phase adjustment (only alignment to approach) is still occasionally required, occurring in approximately 20% of flights, similar to the old tool. Here e.g. a data trained ML algorithm could offer a better performance without the drawbacks of stricter transition criteria. The flight errors detected by the new algorithm are more comprehensible and correspond to the pilot's sense of error during the flight. However also the new tool can currently only be used to analyze docking maneuvers, but it provides a good basis to expand on with new scenarios such as Fly-Around and Rendezvous.

Conclusion

The new Flight Data Evaluation tool, developed in Python, offers enhanced maintainability and flexibility for student and research use while maintaining key functionalities from the old software. The improved visualizations and error detection algorithm provide better insight into docking maneuvers, although some phase transitions still require manual adjustments. Despite these limitations, the new tool offers a solid foundation for future expansion into additional flight scenarios.

Future development of the tool could include the integration of the grading formulas for the Masters seminar, or a database of flights to compare personal performance with others and to evaluate different flight strategies.

Literature
[1] Bosch Bruguera M, Schöneich V, Fink A*, et al.* Spacecraft piloting performance assessment – A computational evaluation methodology for the SIMSKILL Experiment 2017.
[2] Detrell G. SM-Lecture 1: Soyuz Simulator Basics. Munich.
[3] Yu.A. Gagarin Cosmonaut's Training Center. Soyuz Crew Operations Manual (SoyCOM): (ROP-19) (Final) 1998.