SSC19-III-06

Improvements in attitude determination and control of the small satellite Flying Laptop

Steffen Gaisser, Ulrich Mohr, Jonas Keim, Daniel Triloff, Sabine Klinkner
University of Stuttgart Institute of Space Systems
Pfaffenwaldring 29, 70569 Stuttgart, Germany;+49 711 685-62512
gaisser@irs.uni-stuttgart.de

ABSTRACT

Precise attitude control is a key factor of many payloads with high ground resolutions, small fields of view or narrow beams such as an optical data downlink. The small satellite Flying Laptop (FLP), launched in July 2017, was developed by graduate and undergraduate students at the Institute of Space Systems of the University of Stuttgart with support by the space industry and research institutions. The satellite is three-axis stabilized with reaction wheels as main actuators. FLP is equipped with the OSIRIS optical data downlink which was built by the German Aerospace Center (DLR). As this instrument is body mounted on an optical bench, the attitude determination and control system (ACS) is required to point the whole satellite in the direction of the ground station with a high pointing accuracy of 150 arcseconds. At the time of launch the ACS did not reach this precision. This paper describes how the attitude determination and control were improved to achieve the required performance.

The improvements can be divided into two parts. The first part includes the enhancement of on-board sensor processing and attitude control. In the second part, in-orbit data were utilized to increase the accuracy of parameters which are used to control the spacecraft. The first part includes the addition of a Kalman filter, an improved position propagation, and the introduction of adaptive gains to the on-board ACS. The FLP simulation test bed was used to verify the changes. The test bed was also used to find adequate initial values for the Kalman filter and to find inaccuracies in the sensor processing. In the second part, the adaptive gains and the Kalman initial values were validated in-orbit after the upload of the new sensor processing. Moreover, the on-board component orientation settings were corrected for the star trackers, the multi-spectral camera system, and the OSIRIS instrument on FLP.

As a result, the satellite fulfills the pointing requirement of less than 150 arcsecond deviation from the target attitude for a sufficient period of time during a pass over the target. Successful links with the optical data downlink were demonstrated with the DLR ground station in Oberpfaffenhofen.

INTRODUCTION

The small satellite Flying Laptop (FLP), launched in July 2017, was developed and built by graduate and undergraduate students at the Institute of Space Systems of the University of Stuttgart with support by the space industry and research institutions. The mission goals are technology demonstration, Earth observation and education. FLP has a mass of 110kg. The satellite bus architecture is designed to be single fault tolerant. It also features three-axis stabilized attitude control.[1] The payload instruments on FLP are the Multi-spectral Imaging Camera System (MICS), the Panoramic Camera System (PAMCAM), an AIS receiver and the OSIRIS optical data downlink system. Figure 1 shows the satellite without its multi layer insulation before shipping. The MICS system can be seen in the left top module and the OSIRIS collimators are located in the right top module above the data downlink antenna.

FLP finished its LEOP and Commissioning in September 2017 and is operational since then. The on-board software was updated with the changes described in this paper in June 2018 and the first flash from OSIRIS was seen in August 2018.

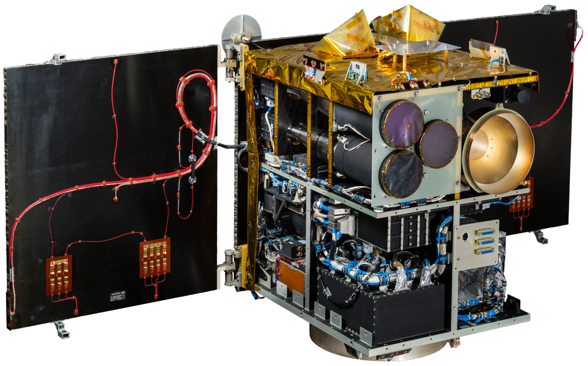

Figure 1: Small Satellite *Flying Laptop*

Attitude Determination and Control

The attitude control system (ACS) uses two different kinds of actuators: Four reaction wheels in a tetrahedral arrangement and three internal redundant magnetic torquers. In addition, the system uses two three-axis magnetometers and eight redundant sun sensors for safe mode attitude determination. Inertial attitude determination in higher modes is accomplished by a star tracker (STR) and four fibre-optic gyros (FOG).

The micro Advanced Stellar Compass (microASC) from the Technical University of Denmark is used for star tracking. This instrument has two camera head units (CHU) to limit blindings during target overflights and to reduce the overall attitude noise around the boresight axis. The performance of the system is specified with an accuracy of $7''$ ($1\ \sigma$) around the boresight of a single camera unit. The camera head sensor temperature is cooled by a radiator placed between the heads. Both cameras deliver a attitude solution with 2 Hz in the current configuration.

The attitude control loop is executed at 5 Hz. It is part of the on-board software, which was developed in-house in cooperation with Airbus.

The attitude control system provides different control modes. The safe mode is the first mode the satellite enters after the boot of the onboard computer. This mode is using the magnetometers and sun sensors as input and keeps the solar panels aligned to the sun with the magentic torquers. The next mode in the mode hierarchy is the idle mode, in which all available sensor data are processed and it controls the attitude based on either sun sensor inputs or a sun model based on STR data. The target attitude is similar to the safe mode but the reaction wheels are used as actuators and magentic torquers are used for wheel desaturation. In addition, the higher attitude modes are completed by the inertial pointing, the nadir pointing and the target pointing. All of these modes use a quaternion feedback controller, which controls the error quaternion and the rate. The difference in-between the modes is the target navigation. Inertial pointing will point the satellite towards an arbitrary inertial attitude. In the nadir pointing mode the satellite aims its instruments towards the Earth's center and in Target pointing mode towards a position on Earth. For the target pointing mode the required pointing accuracy is $150''$. The pointing error is defined as the angle between the target quaternion and the current attitude quaternion. This can be seen in Figure 2. The target quaternion describes a rotation from the inertial reference coordinate system to the target coordinate system. The target coordinate system has its z-Axis pointing towards the target, its x-Axis perpendicular to the orbit normal and the y axis to complete a cartesian coordinate system. Although, the angular error describes a three dimensional angle in the requirement, a stable optical link can be achieved if the angle across the instrument axis of OSIRIS is below the size of the spread of the instrument of $200''$. However, due to alignment uncertainties the error angle around the instrument axis should be low as well.

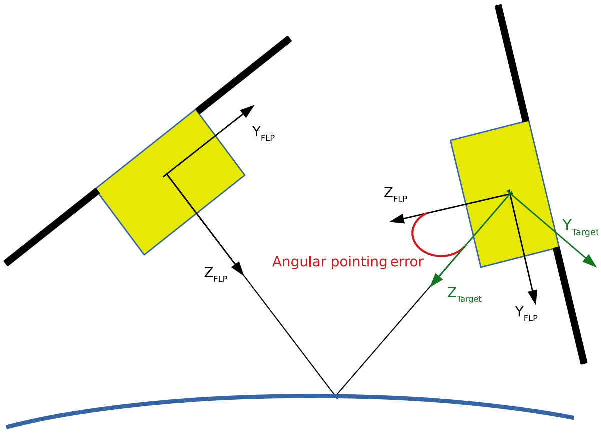

Figure 2: Target Pointing with pointing error between the Target and the FLP system

METHODOLOGY

Due to the constraints of ground based precise attitude tests the ground database of sensor data was limited. Hence, in-orbit data had to be used to determine the sensor performance. The software additions that were developed after launch were verified at the software test bed (STB). This facility is able to simulate all satellite devices in real-time and the onboard computer is connected as a hardware-in-the-loop setup for software tests. In addition, this allows simulations of the complete attitude control system with realistic sensor data delays and protocols. The S/C data were analysed with a NASA Naif SPICE based python tool written at the Institute of Space Systems. The end-to-end test of pointing performance was evaluated with image data from the MICS.

ATTITUDE DETERMINATION

This section describes the attitude determination at the time of launch and the implemented improvements. At the time of launch, the work described in this paper was in preparation.

Status at launch

The approach on the fusion of the quaternion and rate information, which was part of the onboard software at launch, was based on unfiltered sensor data fusion. If the STR provided a solution the software calculated the rotation in relation to the last solution. This rotation rate was used to extrapolate the attitude to compensate the reaction wheel torque delay. If the rate measurement from the FOG was valid as well, this rate was used instead of the quaternion rotation. The rates were used without any further filtering. As a result, the inputs for the attitude controller varied in quality depending on the availability of the STR solutions and the rather high noise level on the FOG. In addition, the fusion of the solutions of both STR heads did not work as intended. The reason was an unexpected delay in-between the solutions of the STR attitude. Ground tests did not show this behaviour because only one of the two camera head units could be used for end-to-end tests with a star simulator. It was expected that both attitude solutions will be received in one telemetry packet or at least in the same cycle of the attitude control system. In contrast to that, first tests during LEOP showed that the solutions are received with a delay in-between. The algorithm that fused the attitude quaternion would only use the solutions in the same time-step. Therefore, no fusion was executed.

Algorithm Improvements

The major change is the implementation of a Kalman filter, which is used to replace the sensor fusion and to improve the accuracy in case of missing sensor data. The filter is an extended Kalman filter (EKF) with a compensation for the delayed STR solutions. The algorithm was developed in earlier stages of FLP and implemented in MATLAB SIMULINK. In first versions of the onboard software an import of code from MATLAB was tested but did not show acceptable execution times. Therefore, a new implementation in *C++* was developed. In contrast to the earlier version, this filter uses the knowledge gained from in-orbit data. The approach is a standard EKF with a virtual quaternion measurement in the Jacobian measurement matrix to fulfill the unit quaternion constraint. The state x_k at time k is defined as:

$$\boldsymbol{x_k} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \\ q_x \\ q_y \\ q_z \\ q_w \end{bmatrix} \tag{1}$$

Where ω_i is the rotation rate in the specific axis and q_i is the quaternion part of the attitude quaternion $\boldsymbol{q}$, where q_w is the real part. The approach is a standard EKF:

$$\boldsymbol{\hat{x}_k} = f(\boldsymbol{\hat{x}_{k-1}}, \boldsymbol{u_{k-1}}) \tag{2}$$

$$\hat{P}_k = F_k P_{k-1} F_k^T + Q_k \tag{3}$$

Where F_k is the Jacobian Matrix of the state transition function and Q_k is the process noise matrix. The error vector is defined as:

$$\boldsymbol{e_k} = \boldsymbol{z_k} - h(\boldsymbol{\hat{x}_k}) \tag{4}$$

Where $\boldsymbol{z_k}$ is the measurement vector and $h(x)$ is the non-linear measurement function. $\boldsymbol{\hat{x}_k}$ is the estimated state vector at time k. The measurement vector is similar to the state vector but includes a virtual measurement of the length of the quaternion:

$$\boldsymbol{z_k} = \begin{bmatrix} \boldsymbol{\omega_{FOG}} \\ \boldsymbol{q_a} \\ \boldsymbol{q_b} \\ 1 \end{bmatrix} \tag{5}$$

To fulfill the unit quaternion constraint with Equation 5 in Equation 4, the function $h(\boldsymbol{\hat{x}_k})$ is defined as:

$$h(\boldsymbol{\hat{x}_k}) = \begin{bmatrix} \boldsymbol{\hat{\omega}_{FOG}} \\ \boldsymbol{\hat{q}_a} \\ \boldsymbol{\hat{q}_b} \\ |\boldsymbol{\hat{q}}| \end{bmatrix} \tag{6}$$

Note that due to the STR delay, the estimations $\boldsymbol{\hat{q}_a}$ and $\boldsymbol{\hat{q}_b}$ correspond to $\boldsymbol{q_a}$ and $\boldsymbol{q_b}$ at the time of the measurement. This is accomplished by interpolating the estimated states to the time of the measurement. Therefore, the Filter saves the last 15 state vectors. This solution is visualized in Figure 3.

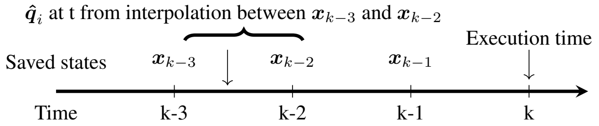
$\boldsymbol{\hat{q}}_i$ at t from interpolation between $\boldsymbol{x}_{k-3}$ and $\boldsymbol{x}_{k-2}$
Execution time
Saved states
$\boldsymbol{x}_{k-3}$
$\boldsymbol{x}_{k-2}$
$\boldsymbol{x}_{k-1}$
Time
k-3
k-2
k-1
k

Figure 3: STR estimation at measurement time t at execution time k

The Kalman Gain is defined as:

$$K_k = \hat{P}_k H_k^T (H_k \hat{P}_k H_k^T + R_k)^{-1} \tag{7}$$

Where H_k is the Jacobian Matrix of the measurement function with 13 × 7 entries. The correction step is then:

$$\boldsymbol{x_k} = \boldsymbol{\hat{x}_k} + K_k \boldsymbol{e_k} \tag{8}$$

$$P_k = (I - K_k H_k)\hat{P}_k \tag{9}$$

The measurement noise covariance matrix R_k of Equation 7 also contains the virtual noise on the unit constraint:

$$R_k = \left[\begin{array}{c|c|c|c} \boldsymbol{\nu_{FOG}} & 0 & 0 & 0 \\ 0 & \boldsymbol{\nu_{qA}} & 0 & 0 \\ 0 & 0 & \boldsymbol{\nu_{qB}} & 0 \\ 0 & 0 & 0 & \nu_{qu} \end{array}\right]_{13\times13} \tag{10}$$

The quaternion noise is modeled as function of the camera head temperature and the spacecraft rotation. However, the MATLAB simulation showed that the accuracy with the model was lower than with a static value for the noise. Additionally, during simulation on the verification test bed the dynamic noise induced oscillations in the filter.

Processing Improvements

In addition to the Kalman filter, the attitude determination and navigation is improved as smaller inaccuracies are fixed. During tests of the navigation a spike in the calculation of the target system rotation was found. The issue could be tracked down to be caused by inter-process communication. The attitude determination runs in a separate process in parallel with the device handlers (DH). These device handlers are objects, which act as software counterparts to the hardware. The state and health management of the device, as well as the communication with the device, are part of their functionality. If the updated information from the device is available, it will be written to the datapool. The datapool is a storage for telemetry, where only one instance is allowed to write and n others are allowed to read. A part of the attitude determination and control is the navigation which calculates the actual target system based on the monitored GPS position. This monitoring also is part of the attitude controller process and is executed before the navigation. Although, the position was checked in the GPS monitoring and written to a variable, which was used later on in the navigation, the time of the GPS position was not written to a new variable. The time was used from the same datapool member, in which the GPS DH writes the new updated time in its cycle. As a result, the navigation calculates a target rate which is higher than expected. This issue is visualized in Figure 4.

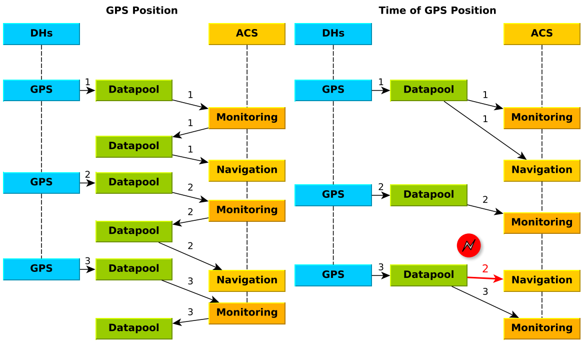

Figure 4: Time of GPS Position issue in step 2

Furthermore, the navigation part showed another inaccuracy. The target system is calculated based on the current target, position, and velocity vector. In the target system one axis is aligned perpendicular to the orbit normal and the target vector, where the orbit normal is the cross product of the position and velocity vector. If the first two vectors are not perpendicular, the result is not a unit vector. As a consequence, the target system shows a jump at the point where both are nearly parallel. Figure 5 shows the calculated three dimensional pointing error of a target pointing with the software version of the launch. The spike in the target system can be seen in the middle of the target pointing. The noise on the rotation information from the same overflight can be seen in Figure 6. The magnitude of the noise is in the order of magnitude of the rotation for the two lower axis.

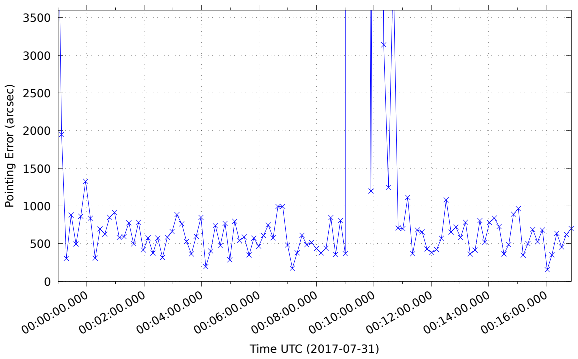

Figure 5: Typical Pointing Error during target pointing with the software version of the launch

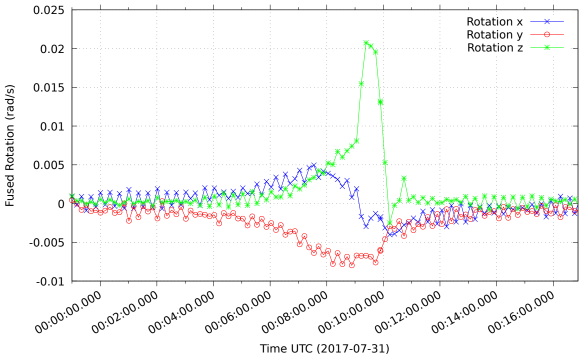

Figure 6: Rotation measurement in the same overflight

The navigation also included two position propagation methods: a geometrical method for short term propagation and a SGP4 propagator for two line element (TLE) based propagation for long term calculation. The short term propagation described the satellite movement on a circular segment and used the velocity vector of the last GPS position to extrapolate the current position and velocity. This method is simple to compute as its based on a few vector operations. However, the velocity propagation was inaccurate after few seconds. Although this method is only used for a few minutes, the difference between the GPS measured position and this method is too large for an accurate target pointing. This can be seen in Figure 7.

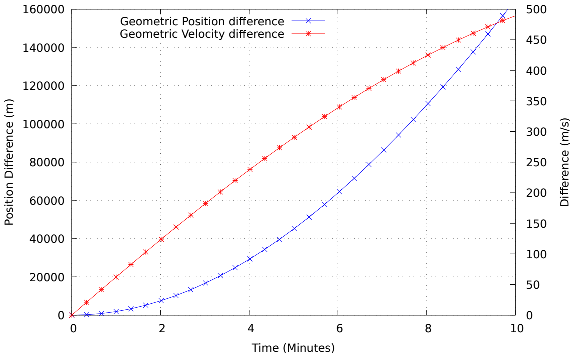

Figure 7: Geometric position extrapolation difference to GPS data from FLP

Therefore, a new method had to be found. The chosen solution is based on the Joint-Gravitational-Model 3 as described by O.Montenbruck.[2] The model uses the JGM3 Model and a fourth order Runge-Kutta method to solve the differential equation.

Parameter Improvements

The software development is one part of the update. The other part consists of the optimization of the parameters for the ACS. First, the sensor mounting of the STR camera head units is measured. Second, the relative attitude of the camera system and optical downlink to the STR is determined. The first task is accomplished by comparing the two solutions of both heads. The inter-boresight-angle (IBA) can be calculated directly from the two quaternions. A special procedure is used to decrease the noise on the sensors and to guarantee that both heads are not blinded during the measurement. During the procedure the satellite points inertial to minimise the rotation rate and, as a consequence, the noise. The target quaternion and time is specified to fulfill the second requirement of keeping both sensor heads out of the exclusion angles around the Earth, the Sun, and the Moon. During the maneuver, the satellite records the STR telemetry with its measuring frequency of 2 Hz. The expected IBA is 56° from the CAD model of FLP. The result of this procedure shows a median angle between the CHU of 55.61°. The corresponding mounting quaternions are then optimized to fit to the camera orientation. Therefore, one STR quaternion is seen as fixed and the other is optimized to minimize the error in-between the solutions. This quaternion is calculated through quaternion weighted averaging according to F. Markley et al.[3] Those weights are calculated by multiplying the inverse of the residual error, which the STR reports, and the number of Stars used for solution.

For the second task, the determination of the relative mountings of the camera system and the optical data downlink to the STR heads, two different approaches are needed. The MICS mounting can be determined direclty from picture data. In contrast to the camera system the OSIRIS mounting matrix has to be determined with data from the ground station. The initial parameter of the mounting matrix was calculated from an experiment in the clean room. The OSIRIS spot was reflected on a wall of the clean room and a picture with the PAMCAM was taken. Although the resolution is lower and the FOV is larger, the PAMCAM was chosen for this experiment because of its mounting position close to the OSIRIS collimators. Starting from this orientation the Laser was spotted during search patterns around this initial attitude.

Another parameter which is based on CAD modelling is the moment of inertia. The experiment to determine its composition could not be completed for this paper. The planed method is based on A.Kornienko et al.[4]

VERIFICATION AND VALIDATION

The verification of the software is an important but difficult task if no engineering model is available. At the Institute of Space Systems a software verification facility (SVF) is used to accomplish this task. The SVF consists of a system test bed (STB) which is based on a hardware-in-the-loop design. The UT699 engineering model of the onboard core board is connected as controller-in-the-loop with a real time simulator. All device communications, the spacecraft dynamics, and the orbital propagation are simulated as well as the thermal state.

The Kalman design was verified with a MATLAB SIMULINK simulation developed at Institute of Space Systems. In this mode only the satellite dynamics and the sensor noise are modeled. However, this simulation was developed in earlier stages of the project. It underestimated the measurement delays of the star tracker and the attitude determination frequency was at 10 Hz instead of 5 Hz. Therefore, the performance was expected to be better than the current version in orbit but it is still useful to show that the design for the filter is correct. The result of the MATLAB simulation showed that the filter is able to provide attitude determination with a three-dimensional angular error of $5''$ to $15''$ during a target pointing pass. With this confirmation of the design, the *C++* code was chosen to be a re-implementation of the MATLAB code. The design changes are that the matrix inversion of Equation 7 is based on Cholesky Decomposition in *C++* and the larger delay is handled as described in Figure 3. In addition, the management of output validity is changed to comply with the flight software standards. The integration of events and monitors is also part of the portation. The verification was done in two steps. At first, the code was run in a standalone version and test data was used to verify the implementation. In the second part, the Kalman filter is embedded in the flight software and tested on the UT699 Board in the STB. The first step showed that the code is correct but the standalone version did not use the interfaces of the final implementation. The second step was able to determine the correctness and delivered estimations of the final performance. Moreover, the STB was used to simulate the operational usage of the filter and to develop procedures for the monitoring during operations. The results at the STB were gathered in several hours of tests in different pointing modes. These tests showed that the performance is, as expected, lower than in the MATLAB model. The knowledge error during the simulations was in the range of $10''$ to $30''$ during a target pointing. The simulations also revealed that the initialization of the filter has to be changed due to the new delay method and to reduce the time until a stable output is reached. The initial quaternion was changed to an evenly distributed quaternion with entries in all 4 dimensions which showed beneficial behaviour. The new delay compensation needed to be accommodated in the initialization as well. The reason is that after a reset of the filter a rotation measurement is available before the STR measurement, or at least at the same time. The initial quaternion will be propagated with this rotation information and saved in the saved states. If a solution of the STR is available at the time k the filter would calculate a phantom rotation between $\boldsymbol{q}_{k-1}$ and $\boldsymbol{q}_k$ which induces an oscillation in the filter. Therefore, at the time k the saved states will be corrected by multiplying the rotated initial quaternion of the saved state by the error quaternion to minimize the phantom rotation. Although, this reduces the amplitude of the oscillations during the initialization it was found that the filter needs to set its output to invalid during the first phase. Otherwise the controller is not able to damp the oscillation in an acceptable time which puts more stress on the reaction wheels. The logic to monitor the validity of the filtering compares the sensor measurements to the final result of fused states. If the fused data are in a reasonable derivation then a confirmation counter is reduced. When this counter reaches a settable limit the output is set to valid. This proved to be a robust solution for the initialization phase. However, the confirmation counter is not used if the filter derivation is too high in a correction step. This causes the filter to set its output to invalid immediately. This solution showed an acceptable amount of occurrences in the simulation which triggered in case of high rotational rates. However, in the operational scenario on FLP it triggered more often. This is related to a transient phenomena of the STR in which a valid solution is received but the angle between the heads show that the solution seems to be wrong. At the moment, there is no onboard process to mitigate this issue. An example of the STR solution jumps can be seen in Figure 8. This might be caused by stray light.

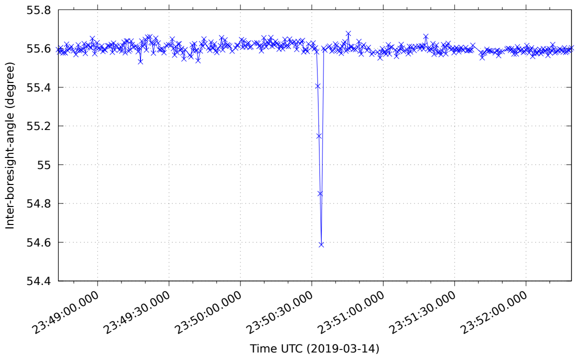

Figure 8: Inter-boresight-angle with jump

The noise filtering performance of the Kalman filter can be seen Figure 9. The noise on the fused rotation is in the expected magnitude in comparison with Figure 6. Note that the sampling frequency is higher than in Figure 6. At the end of the pass in Figure 9 a reaction wheel zero crossing induced a low amplitude oscillation in the controller.

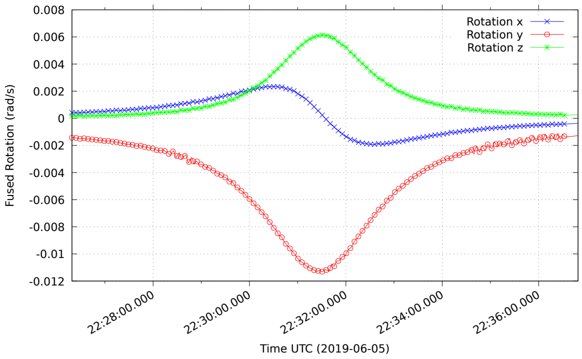

Figure 9: Rotation rate with Kalman filter during overflight

In addition to the filter, the processing issues mentioned above were found at the STB during the tests of the Kalman filter. Their respective fixes were also tested at the STB. The position propagation described above was validated by comparing it to measured in-orbit data from FLP. The results are shown in Figure 10. With the new propagation it is possible to propagate a full overflight of 10 minutes without larger deviations compared to 7.

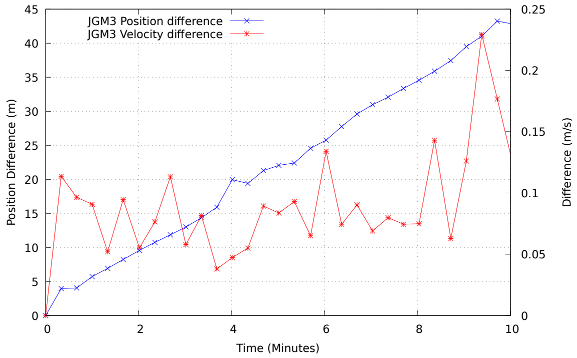

Figure 10: JGM3 position extrapolation difference to GPS data from FLP

The validation of the in-orbit performance is more challenging because of the reduced amount of available telemetry. The pointing performance analysis is done with a tool which uses the python wrapper SpiceyPy for the NASA NAIF SPICE toolkit. This tool can be used to import the telemetry of FLP and to calculate the expected pointing direction in the spacecraft frame. The tool has been developed during a master thesis at the University of Stuttgart in cooperation with ESA.[5] The model does not use a height model of the earth at the moment. This limits the precision for places with larger distance to the WGS84 ellipsoid. However, the tool proved to be useful for the calculation of the CHU mounting matrix relative to the instruments. The process flow for the mounting matrix can be seen in Figure 11.

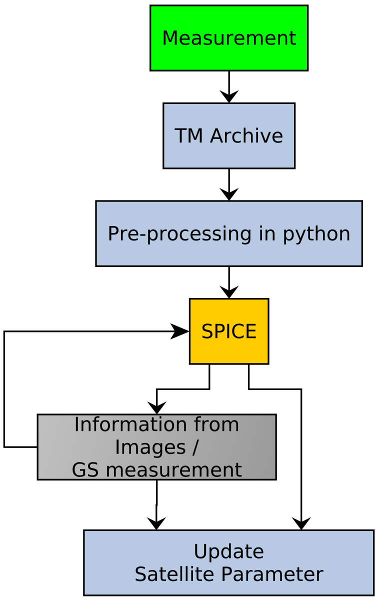
Measurement
TM Archive
Pre-processing in python
SPICE
Information from Images / GS measurement
Update Satellite Parameter

Figure 11: Process of data processing

In case of a picture from the MICS the measurement of pointing data will be preprocessed in a python toolchain. The task of preprocessing is the calculation of raw data from the STR as well as the preparation of the correct file format for the SPICE tool. The tool expects STR quaternions to rotated in the body system. The first processing step in the SPICE tool is the calculation of the vector to the target point on the mid-pixel of the MICS at the time of the picture. The second step is the calculation of a corresponding quaternion from the vector. This aligns the body system z-axis with the instrument axis. This quaternion can be used to rotate the mounting matrix of the two CHU. Afterwards, the calculation can be verified by computing the expected intersection of the instrument axis with the earth in SPICE and compare it with the actual picture. If both align inside the expected margin the system parameter can be updated. This margin is caused by the time-variant knowledge error. The result of this method is a more precise MICS pointing which is validated by taking pictures of distinctive places on earth and comparing the angular distance of the mid-pixel to the target. The angular distance is calculated with an openCV based python script, which uses a picture with colored pixels as input. With this method the end-to-end pointing error of the satellite system can be measured including the uncertainties in the mounting matrix. Figure 12 shows the results for a target overflight with a maximum elevation of 83° above the target at 06:38:16.

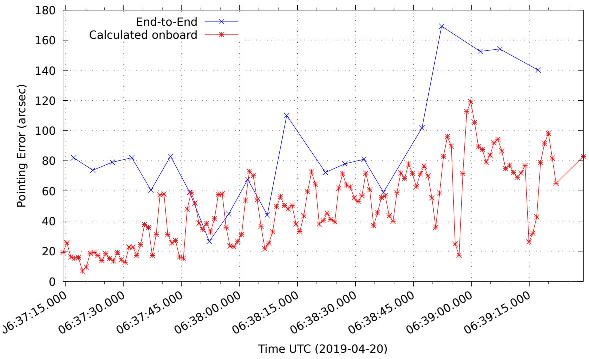

Figure 12: End-to-End target pointing performance from MICS pictures during high elevation overflight

The top curve shows the end-to-end error and the bottom curve the calculated pointing error. The end-to-end pointing error is measured by the angular distance of the target to the MICS mid-pixel. Notice that this corresponds to the 2 D angle projection of the 3 D pointing error. This means that this is the pointing error across the instrument axis. The time variant deviation between the two curves is caused by the knowledge error of the satellite. The time-invariant offset between the curves is equivalent to the mounting uncertainty. The clear oscillation overlaying the onboard data is caused by a reaction wheel zero crossing. This can not be seen in the end-to-end data because of the low measuring frequency. With respect to the OSIRIS instrument, which is required to point its boresight inside the beam divergence of $200''$, this performance is sufficient for the system.

In contrast to the MICS, the OSIRIS direction can not be verified with a picture. However, as the SPICE tool method showed sufficient performance in the post processing of the MICS, it is also applied to the OSIRIS mounting matrix. In this case the input is the received signal power on the ground station. SPICE is then used to calculate the vector from FLP to the GS at any moment during a overflight. The mean value is determined with a weighting of the received power and the inverse of the distance. As a result multiple successful OSIRIS links were demonstrated with the Institute of Communications and Navigation of the German Aerospace Center which shows the repeatability of the target pointing.

CONCLUSION

The *Flying Laptop* mission was not able to fulfill its pointing requirements for the target pointing mode at the time of launch. With the improvements described in the paper, the satellite is able to fulfill the requirement for earth pointing with OSIRIS, the optical data downlink system. The addition of a Kalman filter resulted in a significant lower noise level on fused sensor data. Moreover, other smaller inaccuracies in the navigation were fixed and onboard parameters were determined. A toolchain to analyse the satellite data was developed and used to determine the FLPs geometrical parameters.

Further work will be done on the filtering of flickering star tracker data and the measurement of the moment of inertia to improve the attitude determination and control.

References

[1] Eickhoff, J., *The FLP Microsatellite Platform: Flight Operations Manual*, Springer Aerospace Technology, Springer International Publishing, 2015.

[2] Montenbruck, O. and Gill, E., *Satellite Orbits: Models, Methods, and Applications*, Physics and astronomy online library, Springer Berlin Heidelberg, 2000.

[3] Markley, F. L., Cheng, Y., Crassidis, J. L., and Oshman, Y., “Averaging Quaternions,” *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 4, 2007, pp. 1193–1197.

[4] Kornienko, A., Dhole, P., Geshnizjani, R., Jamparueang, P., and Fichter, W., “Determining Spacecraft Moment of Inertia Using In-Orbit Data,” 06 2017.

[5] Grass, M., *Developement of an open-source Attitude and Orbit Simulation Tool*, Master’s thesis, University of Stuttgart and European Space Astronomy Center Data and Engineering Division, 7 2018.