Available online at www.sciencedirect.com

ScienceDirect

Advances in Space Research xxx (xxxx) xxx

ADVANCES IN SPACE RESEARCH (a COSPAR publication)

www.elsevier.com/locate/asr

A robust navigation filter fusing delayed measurements from multiple sensors and its application to spacecraft rendezvous

Heike Frei *, Matthias Burri, Florian Rems, Eicke-Alexander Risse

German Aerospace Center (DLR), Muenchner Str. 20, 82234 Wessling, Germany

Received 3 March 2022; received in revised form 4 October 2022; accepted 7 October 2022

Abstract

A filter is an essential part of many control systems. For example guidance, navigation and control systems for spacecraft rendezvous require a robust navigation filter that generates estimates of the state in a smooth and stable way. This is important for a safe spacecraft navigation within rendezvous missions. Delayed, asynchronous measurements from possibly different sensors require a new filter technique which can handle these different challenges. A new method is developed which is based on an Extended Kalman Filter with several adaptations in the prediction and correction step. Two key aspects are extrapolation of delayed measurements and sensor fusion in the filter correction. The new filter technique is applied on different close-range rendezvous examples and tested at the hardware-in-the-loop facility EPOS 2.0 (European Proximity Operations Simulator) with two different rendezvous sensors. Even with realistic delays by using an ARM-based on-board computer in the hardware-in-the-loop tests the filter is able to provide accurate, stable and smooth state estimates in all test scenarios.

Keywords: Navigation filter; Delayed measurements; Sensor fusion; Hardware-in-the-loop test; Rendezvous

1. Introduction

1.1. Background

A filter estimates a time-varying state using a model of the system dynamics in combination with measurements collected at discrete times. The measurements are related to the state via some measurement model. Both system model and measurement model are uncertain, and the measurements are afflicted with noise. In control loops, one uses the state estimation and compares it with a desired value. Often pure measurements are very noisy and cannot be used directly. Sometimes not all components of the state can be measured directly. Therefore, filter methods are applied to combine knowledge about the underlying system dynamics with measurements (Simon, 2006; Zarchan and Musoff, 2000).

Estimation problems have to be solved in many different disciplines and applications. One application is the control of aerospace vehicles and spacecrafts (satellites, space stations, supply vehicles, etc.). Examples are attitude determination and control of spacecrafts (Wertz, 2002; Zivan and Choukroun, 2021), orbit control during automated ren-

Abbreviations: GNC, Guidance, Navigation and Control; EPOS, European Proximity Operations Simulator; EKF, Extended Kalman Filter; L-VLH, Local Vertical Local Horizontal; ECI, Earth Centered Inertial; LiDAR, Light Detection and Ranging; PMD, Photonic Mixer Device; O-BC, On-Board Computer; HiL, Hardware-in-the-Loop

* Corresponding author.

E-mail addresses: heike.frei@dlr.de (H. Frei), matthias.burri@dlr.de (M. Burri), florian.rems@dlr.de (F. Rems), eicke.risse@web.de (E.-A. Risse).

https://doi.org/10.1016/j.asr.2022.10.025
0273-1177/

dezvous and docking (Fehse, 2003), and on-board navigation (orbit/attitude/trajectory estimation) in manned and unmanned spaceflight (Grewal and Andrews, 2010; Volpe et al., 2022). Typical navigation sensors are inertial sensors, camera systems (for example star trackers or rendezvous cameras), Radio Frequency (RF) sensors, Light Detection and Ranging (LiDAR) systems, and Global Navigation Satellite System (GNSS) receivers (Fehse, 2003; Grewal and Andrews, 2010; Marchand et al., 2019; Persson et al., 2006; Volpe et al., 2017; Volpe et al., 2022; Wertz, 2002; Zhang et al., 2005b; Zivan and Choukroun, 2021).

1.2. Motivation and problem formulation

For state estimation, we consider the following discrete-time system:

$$\vec{x}_{k+1} = f\left(\vec{x}_k, \vec{u}_k, t_k\right) + \vec{v}_k, \tag{1a}$$

$$\vec{z}_k = h\left(\vec{x}_k, t_k\right) + \vec{\omega}_k, \tag{1b}$$

where $\vec{x}_k \in \mathbb{R}^n$ denotes the state vector at time $t_k \in [0, \infty)$, $\vec{u}_k \in \mathbb{R}^l$ the input or control vector, $\vec{v}_k \in \mathbb{R}^n$ the system noise modelled as additive noise, $\vec{z}_k \in \mathbb{R}^m$ the measurement and $\vec{\omega}_k \in \mathbb{R}^m$ the measurement noise.

Further $f : \mathbb{R}^n \times \mathbb{R}^l \times [0, \infty) \to \mathbb{R}^n$ is a possibly non-linear function which models how the new state depends on the previous state, on a control vector and on time. f is called system model. The function $h : \mathbb{R}^n \times [0, \infty) \to \mathbb{R}^m$ is called measurement model. It describes how the measurement depends on the state vector and on time.

In some applications, only part of the state vector can be measured. In those cases the measurement model is a reduction of the state vector to a vector containing those components which can be measured. For example, the state vector could consist of a 3D position $\vec{p}_k \in \mathbb{R}^3$ and a 3D velocity $\vec{v}_k \in \mathbb{R}^3$ and the measurement is the position only. Then the state vector can be written as $\vec{x}_k = \left(\vec{p}_k, \vec{v}_k\right)^T \in \mathbb{R}^6$ and the measurement as $\vec{z}_k = \vec{p}_k \in \mathbb{R}^3$. But, in general, also a non-linear relation between state vector and measurement vector can exist. For example, the measurement of a sensor could be the azimuth and elevation angle and the state vector is a 3D Cartesian position vector. Or the measurement model can represent a time-dependent coordinate transformation. This is often the case, for example when a measurement is given in the sensor frame and the estimate should be determined in another coordinate frame.

The classical Kalman filter, as presented by Kalman (1960), provides a special solution to the linear filter problem. It assumes that f and h are linear and the noise is Gaussian with zero mean and known covariance. The Extended Kalman Filter (EKF) similarly provides a method to estimate the state by linearization around the current state (see for example Zarchan and Musoff (2000) for different filter techniques, incl. the EKF). Many filter techniques use a filter prediction step, where the system dynamics model is used to propagate the state from t_k to t_{k+1} followed by a filter correction step, where the estimation is updated by using the measurement.

In many filter techniques it is assumed that a measurement $\vec{z}_{k+1}$ is known when computing the new state estimate $\vec{x}_{k+1}$. However, this is only an ideal case. In real applications, the measurement is not necessarily given at time t_{k+1}. When the new filter estimate should be computed, a measurement $\vec{z}(t)$ with $t \leqslant t_{k+1}$ is given. $t = t_{k+1}$ is just a rare special case. Often, measurements are produced via some complex data processing. For example, when a camera is used as a sensor, measurements are typically generated via some image processing routine. The time span from the capturing of the image until the time when the final measurement is available for the filter can be several seconds. It typically depends on the communication path, the interfaces between sensor and processing system and on the computer or embedded system which performs the image processing. The time span $\Delta t = t_{k+1} - t$ further need not be a multiple of the filter sample time $t_{k+1} - t_k$. The delays do not have to be constant and similarly, the filter does not have to be executed with fixed sample times.

Another aspect is the size of the measurement vector. We would like the filter to cope with measurements produced by different sources or different sensors. The time for data processing and the generation of measurements can vary a lot and measurements of different sensors are usually asynchronous. There can be sensors and measurement techniques which can provide a measurement in a few milliseconds and other sensors and techniques need a few seconds. It is not sensible to wait until the last measurement of all sensors is ready. When one of the measurements is completed, the filter should use it in its next filter execution step. Thus, there can be steps where no new measurement is available, steps where only a subset of sensors produce measurements, and steps where all sensor measurements are available. The filter should cope with all these cases.

The handling of asynchronous measurements is especially important in multi-threading, exploiting the increased performance of multi-core computers that are becoming (and for advanced applications have to become) more widespread in space.

In summing up, very complex cases and situations can occur. This motivates us to develop a filter technique which is not restricted to special cases as often used in pure theoretical studies. The filter technique should be robust with respect to delayed measurements, non-constant measurement delay, non-constant filter frequency, and measurements from multiple different sources available at different time points.

1.3. State of the art

Existing studies and works considering filter methods with delayed measurements differ in the way how the

delayed measurements are included in the Kalman filter. Our method is mainly based on the method proposed by Larsen et al. (1998), where a delayed measurement is extrapolated to the present time. However the authors consider the special case in which the time of the delayed measurement is synchronized with the filter, i.e. that the measurement is given at some time $t_i, i \in \{1, \ldots, k\}$ and used by the filter at time t_{k+1}. In this paper, we use the main idea of Larsen et al. (1998) and extend it to more complex cases. Further, only a theoretical approach is given by Larsen et al. (1998). No detailed results are presented.

Another approach dealing with delayed measurements is presented by Bar-Shalom (2002). The focus lies in computing the exact solution by a retrodiction of the state from the latest time back to the time of the delayed measurement and taking the process noise into account. In a later work, Zhang and Bar-Shalom (2012) present an advanced version of Bar-Shalom (2002) which deals with out-of-sequence measurements from multiple sensors. Bar-Shalom (2002), Zhang and Bar-Shalom (2012) present theoretical examples but no single example with real sensor data.

Zhang et al. (2005a) consider measurements which are not time-synchronized with the filter execution times ("out-of-sequence" measurements). The authors present a so-called global optimal update, where the measurement residuum is modified to handle the delay. For this paper, we will use a similar change of the measurement residuum and motivate this approach by studying an extrapolation. Results with real measurement data are not presented by Zhang et al. (2005a). Their paper is purely theoretical, with some numerical results. The global optimal update method presented by Zhang et al. (2005a) reduces to the method of Bar-Shalom (2002) when limited information is given.

Larsen et al. (1998), Zhang et al. (2005a) and Bar-Shalom (2002) do not give any implementation details.

A Kalman filter for relative spacecraft position and attitude estimation is presented by Kim et al. (2007) and an improvement was given by Zhang et al. (2014). The Kalman filter is applied to a real space application with vision-based navigation but the delay case is not considered.

As mentioned in Section 1.2 the filter has to handle delayed measurements but also measurements from different sources. So the method should easily be extendable to the case that more than one delayed measurement arrives at one filter update step, i.e. the size of the measurement vector varies.

In literature, sensor fusion is performed in different ways. Tzschichholz et al. (2015) and Klionovska et al. (2018) take the distance component of a position from one sensor, and the lateral component from a second sensor. No filter is used. The information from measurement components not considered with this method may still be valuable but is lost completely. Both measurements have to be available at one and the same time step for fusion. Further, the method cannot be easily extended to the case of more than two sensors.

DeKock et al. (2008) present an example where sensor fusion is performed at Kalman filter level and applied to relative navigation for autonomous rendezvous and docking. Sensor measurements such as RF, LiDAR and vision-based measurements are included in an EKF. The navigation system is implemented with Matlab/Simulink and tested in the Flight Robotics Laboratory of NASA. It is presented how the different measurement vectors are adapted to the different sensor sources for the special application. However more complex cases, with delay and no time-synchronization of the measurements are not considered.

In Benninghoff et al. (2014), we present a first version of an EKF that can handle delayed measurements. However, the case that a measurement can even be older than the last filter update time, is not considered by Benninghoff et al. (2014) and requires an improvement of the method such that also this case is covered. Benninghoff et al. (2014) present a ground-in-the-loop environment and two test cases. The implementation is done with Matlab/Simulink but the filter is not yet implemented for real on-board computer hardware.

1.4. Contribution of the paper

In this paper, we present a navigation filter that incorporates delayed measurements and sensor fusion which works also for a variety of special cases, often not covered in literature: First, the delay need not be a fixed delay. Second, it need not be a multiple of the filter sample time, i.e. it need not be time-synchronized with the filter execution time. Third, the delayed measurement can also be older than the last filter update step, which requires special care. Fourth, the filter can handle delayed measurements from different sensors and performs sensor fusion. The number of sensor measurements available at one filter execution step need not be constant. Fifth, a constant filter execution frequency is not required. All systems (filter, sensors) can be asynchronous.

In contrast to theoretical papers in literature, we embed the navigation filter in a Navigation and Control (GNC) system for close range rendezvous with real physical sensors to prove its applicability and robustness in a Hardware-in-the-Loop (HiL) scenario. Therefore, we demonstrate that the filter works in a robust and accurate way also in presence of real, non-Gaussian, measurement noise. The measurement noise is affected by different sources of disturbances. This is why Gaussian measurement noise cannot be assumed (Zhang, 2000). The filter and the GNC system are tested in a closed-loop environment. The filter result affects thus the real state and the next measurement. It is demonstrated that the entire closed control loop is stable.

In addition, the paper describes how the filter can be implemented in C++, how the filter states from the past can be efficiently stored in buffers and how the sensor

fusion and the available measurements can be managed using two bitmasks.

The paper is structured as follows: We present the theoretical framework in Section 2. It contains a detailed description of the navigation filter that can handle delayed measurements and sensor fusion, and a description of the GNC system. Further some implementation details are given. In Section 3, the test environment and the HiL test facility European Proximity Operations Simulator (EPOS) are described. In Section 4, results of our HiL tests at EPOS are presented before giving a final discussion and conclusion in Section 5.

2. Methods

2.1. State estimation with delayed measurements

In Section 1.2 the time-discrete system (1) and the delay problem are discussed. The method presented in this paper can be applied to all filtering methods which consist of a prediction step, where the system model is used to propagate the state, and of a correction step, where the predicted state is improved by using measurements (measurement update).

Note that the term *measurement* in this paper, refers to an input to the navigation filter which can be a direct output of a sensor but also the output of a data processing algorithm. The latter is often used in cases when the sensor delivers raw data like an image which is processed to obtain a physical quantity like a position, a pose (position and attitude) or similar.

2.1.1. Example

Fig. 1 visualizes an example with two delayed measurements. Let k be an index for counting the discrete time points when a filter estimate is computed, $t_{k-1}, t_k, t_{k+1}, \ldots$ denote the corresponding time points. At time t_{k-1}, there is no new measurement in our example, i.e. the filter cannot include a measurement to generate an estimation, it computes the prediction step only.

At some time $t_i^{meas} \in (t_{k-1}, t_k)$ a new measurement is started. Here i denotes the index for counting measurements. (Note, that i and k are different indices. For example, there can be the filter execution step number $k = 100$, but only $i = 10$ measurements have been taken.).

The sensor which generates measurement data can be for example a camera which takes an image at the time point t_i^{meas}. After the image is captured, the raw image data is used by some image processing routine which needs a certain, not necessarily constant processing time.

Therefore there is a delay between the time of the image capturing and the time when the filter can use the resulting measurement, the image processing result. At time t_k in our example, presented in Fig. 1, this result is not yet available. However at time t_{k+1} the measurement can be used. The time delay in this case is $t_{k+1} - t_i^{meas}$.

The next measurement is taken at t_{i+1}^{meas}. In our example, $t_{i+1}^{meas} \in (t_k, t_{k+1})$. The measurement is related to a state at a time before t_{k+1}. In this example measurement number i is used by the filter after measurement $i+1$ is started. We assume that software implementations of measurement process and filter run in different, parallel tasks. The measurement system starts processing the next image as soon as the last image processing is finished or by any external trigger, not synchronized to the filter. Sample times of the filter may not be taken into account.

The filter uses a measurement as soon as it can. If there is no new measurement, the filter executes the propagation, i.e. the prediction step, only. In the example visualized in Fig. 1, the measurement $i+1$ even has a larger delay compared to the previous measurement. It "arrives" at the filter shortly before t_{k+4} and can be used at time t_{k+4}.

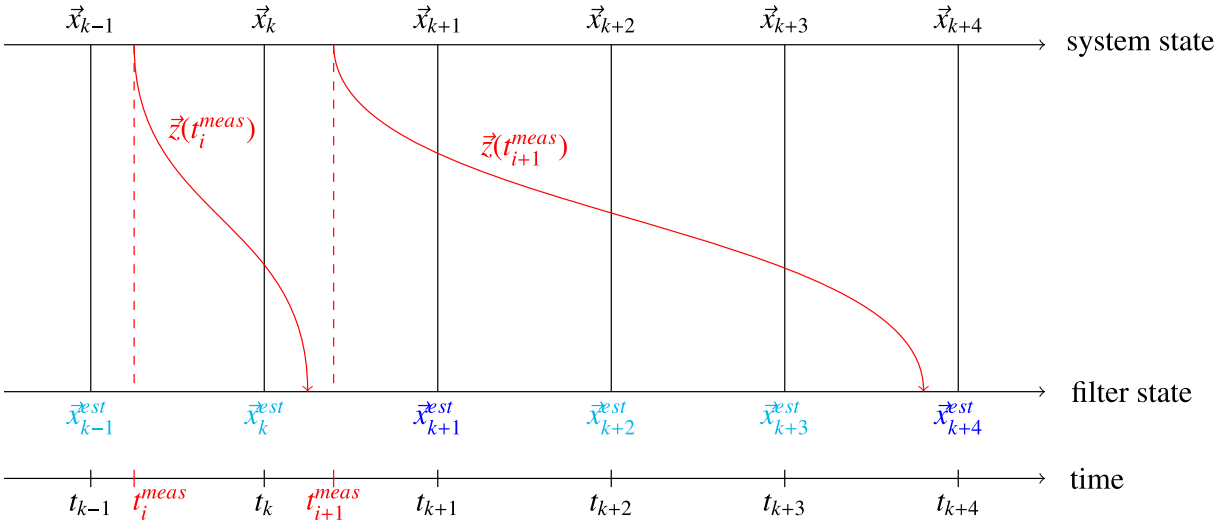


Fig. 1. Illustration of the delay problem (black: actual state and time line, red: measurements and measurement times, light blue: filter estimates by propagation only, blue: filter estimates by propagation and use of measurement), adapted from Larsen et al. (1998).

This example shows a case where the measurement delay is not constant and filter and measurement are asynchronous.

2.1.2. Classical filter

The EKF without delayed measurements computes an estimation of the filter problem (1) as follows: Let $\vec{x}_k^{est}$ be the last filter estimate, i.e. an estimate of the state $\vec{x}$ at time t_k. In the ideal case, if there is no time delay, a measurement $\vec{z}_{k+1}$ of the state at time t_{k+1} is given. The EKF proposes the following algorithm (Zarchan and Musoff, 2000):

$$\vec{x}_{k+1}^{pre} = f\left(\vec{x}_k^{est}, \vec{u}_k, t_k\right), \tag{2a}$$

$$P_{k+1}^{pre} = F_k P_k^{est} F_k^T + Q_k, \tag{2b}$$

$$K_{k+1} = P_{k+1}^{pre} H_{k+1}^T \left(H_{k+1} P_{k+1}^{pre} H_{k+1}^T + R_{k+1}\right)^{-1}, \tag{2c}$$

$$\vec{x}_{k+1}^{est} = \vec{x}_{k+1}^{pre} + K_{k+1}\left(\vec{z}_{k+1} - H_{k+1}\vec{x}_{k+1}^{pre}\right), \tag{2d}$$

$$P_{k+1}^{est} = (\mathbb{I} - K_{k+1}H_{k+1})P_{k+1}^{pre}, \tag{2e}$$

In (2) $\vec{x}_{k+1}^{pre} \in \mathbb{R}^n$ denotes the predicted state vector. It is obtained by using the system model f applied on the previous filter estimation $\vec{x}_k$ and on the control vector $\vec{u}_k$ at time t_k, see sub-Eq. (2a).

The filter estimates both the state and the covariance of the state. $P_{k+1}^{pre} \in \mathbb{R}^{n\times n}$ denotes the predicted covariance, computed as described in (2b). $F_k \in \mathbb{R}^{n\times n}$ is the Jacobian matrix containing the partial derivatives of the system model f with respect to the single components of the state $\vec{x}$ evaluated at time t_k. P_k^{est} is the covariance matrix of the last filter step. The matrix $Q_k \in \mathbb{R}^{n\times n}$ is the covariance matrix of the system noise $\vec{v}_k \in \mathbb{R}^n$.

The matrix $K_{k+1} \in \mathbb{R}^{n\times m}$, a matrix with n rows and m columns, is called filter gain matrix, see Eq. (2c). It uses the predicted covariance matrix and the measurement matrix $H_{k+1} \in \mathbb{R}^{m\times n}$ which contains the partial derivatives of the measurement model h with respect to the state vector $\vec{x}$ evaluated at time t_{k+1}. The matrix $R_{k+1} \in \mathbb{R}^{m\times m}$ is the measurement covariance matrix, i.e. the covariance matrix of the measurement noise $\vec{\omega}_{k+1} \in \mathbb{R}^m$.

The state prediction $\vec{x}_{k+1}^{pre}$ is updated/ corrected using the measurement $\vec{z}_{k+1} \in \mathbb{R}^m$ as described in Eq. (2d). The so-called measurement residuum $\vec{z}_{k+1} - H_{k+1}\vec{x}_{k+1}^{pre}$ is applied on the gain matrix and the result is added to the predicted state vector. Finally the predicted covariance is updated as given in Eq. (2e) resulting in a new state covariance matrix P_{k+1}^{est} (Note that $\mathbb{I}$ denotes the $n \times n$ identity matrix.).

For handling delayed measurements we use as basis the method we proposed in Benninghoff et al. (2014). We develop some modifications, such that more difficult problems can be handled: systems with large velocities require a modification of the filter prediction step; and certain delay cases require a modification of the filter correction step. We describe the method in detail in the next sub-sections.

2.1.3. Filter prediction

Dependent on the application, it can be sufficient for the accuracy, if the filter performs the prediction step with a time step of $t_{k+1} - t_k$. But, since usually the discrete system (2a) is a numerical solution of a differential equation, smaller intermediate time steps are required to increase the accuracy of the predicted state estimate.

We propose to propagate from t_k to t_{k+1} using smaller time steps: Let Δt_{sub} be a desired time step. If $t_{k+1} - t_k < \Delta t_{sub}$ we set $\Delta t_{k,sub} = t_{k+1} - t_k$. Else, we compute the nearest integer below $n_k = \text{floor}\left(\frac{t_{k+1}-t_k}{\Delta t_{sub}}\right)$ and set $\Delta t_{k,sub} = \frac{t_{k+1}-t_k}{n_k}$. (We use a time step size close to Δt_{sub} such that an integer multiple of the sub time step size is $t_{k+1} - t_k$.) To compute $\vec{x}_{k+1}^{pre}$ we iteratively compute n_k times the discrete propagation with time steps $\Delta t = \Delta t_{k,sub}$.

2.1.4. Filter correction

Typically, there can be some time steps, where no new measurement is available. In this case, the filter propagates only, no correction can be made. It computes $\vec{x}_{k+1}^{pre}$ as described above, computes (2b) and sets $\vec{x}_{k+1}^{est} = \vec{x}_{k+1}^{pre}$ and $P_{k+1}^{est} = P_{k+1}^{pre}$, since no correction can be done in absence of new measurements.

If a new measurement $\vec{z}(t^{meas})$ with $t^{meas} \leqslant t_{k+1}$ is available, a measurement residuum is computed. The measurement residuum $\vec{z}_{k+1} - H_{k+1}\vec{x}_{k+1}^{pre}$ (see (2d)) can only be used in the special case $t^{meas} = t_{k+1}$. For the general case, we use the residuum

$$\vec{z}(t^{meas}) - H(t^{meas})\vec{x}^{est}(t^{meas}), \tag{4}$$

where $\vec{x}^{est}(t^{meas})$ is a state estimation and $H(t^{meas})$ is the Jacobian matrix of h with respect to the state evaluated at $\vec{x}^{est}(t^{meas})$ and at time t^{meas} (see also Zhang et al., 2005a). The state estimation $\vec{x}^{est}(t^{meas})$ however is not directly available since the filter is executed at discrete time points $t_0, t_1, \ldots, t_k, t_{k+1}$. However, if a sufficient number of past filter estimates is stored, the estimation $\vec{x}^{est}(t^{meas})$ can be computed using estimates close to t^{meas}. The number of stored estimates is dependent on the assumed maximum delay time plus some margin, see also Section 2.4.

It is therefore important to know the time of the measurement. If one would treat a delayed measurement as instantaneous measurements and if one would compare such a measurement directly with $H_{k+1}\vec{x}_{k+1}^{pre}$ in the measurement residuum like in the classical EKF (see (2d)), we would compute a difference of two quantities related to two completely different time points.

In Benninghoff et al. (2014) we mathematically justified the setting of the measurement residuum (4). It is based on the following idea: Since no real measurement $\vec{z}_{k+1}$ is available, we approximate and replace it with an extrapolated one (Larsen et al., 1998)

$$\vec{z}^{extra}_{k+1} := \vec{z}(t^{meas}) + H_{k+1}\vec{x}^{pre}_{k+1} - H(t^{meas})\vec{x}^{est}(t^{meas}). \quad (5)$$

Eq. (5) can be derived by considering the measurement equation and by performing a Taylor series expansion. For more details, we refer to Benninghoff et al. (2014).

Now we focus on the question how to compute $\vec{x}^{est}(t^{meas})$ including all special cases which can occur. To handle delayed measurements we do not only store $\vec{x}_k$ but also N additional last state estimates (for example using a ring buffer). Thus the estimates $\vec{x}^{est}_{k-N}, \ldots, \vec{x}^{est}_{k-1}, \vec{x}^{est}_{k}$ are available. N should be chosen sufficiently large such that all typical delays can be handled (dependent on the application and the average filter execution frequency).

In the following, we distinguish between different delay cases which can occur in practice:

Case 1. Let $s \in \{k-N, \ldots, k\}$ be an index such that $t^{meas} \in [t_s, t_{s+1}]$ and t_s is newer or equal than the last filter correction time step. In Benninghoff et al. (2014) we proposed to interpolate linearly between known filter estimates $\vec{x}^{est}_s$ and $\vec{x}^{est}_{s+1}$. However the linear interpolation is not accurate enough. Similarly, as for the filter prediction step (Section 2.1.3), we should rather use small propagation steps.

This is visualized in Fig. 2 by a one-dimensional example: The black line represents a non-linear state over time. The light blue line is the filter estimation. Dark blue dots visualize the time point when a filter correction step is executed. Red dots mark measurements. Dashed lines mark projection of points to the time line or state line respectively.

In Fig. 2, the filter is executed for the first time at time t_0 by using an initial guess to start the iterative process. At a time $t^{meas}_1 \in [t_0, t_1]$ a first measurement is started. As discussed above, this can be the time when a measurement source like an image is captured by a sensor. At time t_1 the measurement is not yet available and cannot be used. The filter propagates only. As described in Section 2.1.3, the interval $[t_0, t_1]$ is split into sufficiently small subintervals for the propagation. Here, three intermediate points between t_0 and t_1 are used since the linear propagation with the larger time step size would be too inaccurate. (The time stamp size for the propagation is application-dependent/ is a tuning parameter.) Since no correction can be made in the absence of an available measurement at t_1, the estimation is set to $\vec{x}^{est}_1 = \vec{x}^{pre}_1$.

At time t_2 the measurement $\vec{z}(t^{meas}_1)$ is available and can be used by the filter. The measurement relates to the time t^{meas}_1 which is in the past, but it is the most recent measurement and helps improving the state estimation at time t_2. For computing a measurement residuum, the state estimate $\vec{x}^{est}_0$ is used and via propagation an estimate $\vec{x}^{est}(t^{meas}_1)$ is found. The next measurement $\vec{z}(t^{meas}_2)$ starts in this example in the time interval $[t_2, t_3]$. Now, the delay is less than one filter execution, so in this case, the measurement can be used at time step t_3 and another correction can be done. As before, an estimate $\vec{x}^{est}(t^{meas}_2)$ is computed via propagation.

For case 1 we can formulate in general: for a time step t_{k+1} where a filter update should be computed, we assume that $t^{meas} \in [t_s, t_{s+1}]$, for some $s \leqslant k$ and no filter correction step has been done since t_s. Case 1 occurs in practice, when measurement data is captured, processed and the result is used by the filter in the correction step, and only after this step the next measurement data is captured. If the precondition, that no filter correction step has been done in the meantime, is not satisfied, then this situation is handled by case 2.

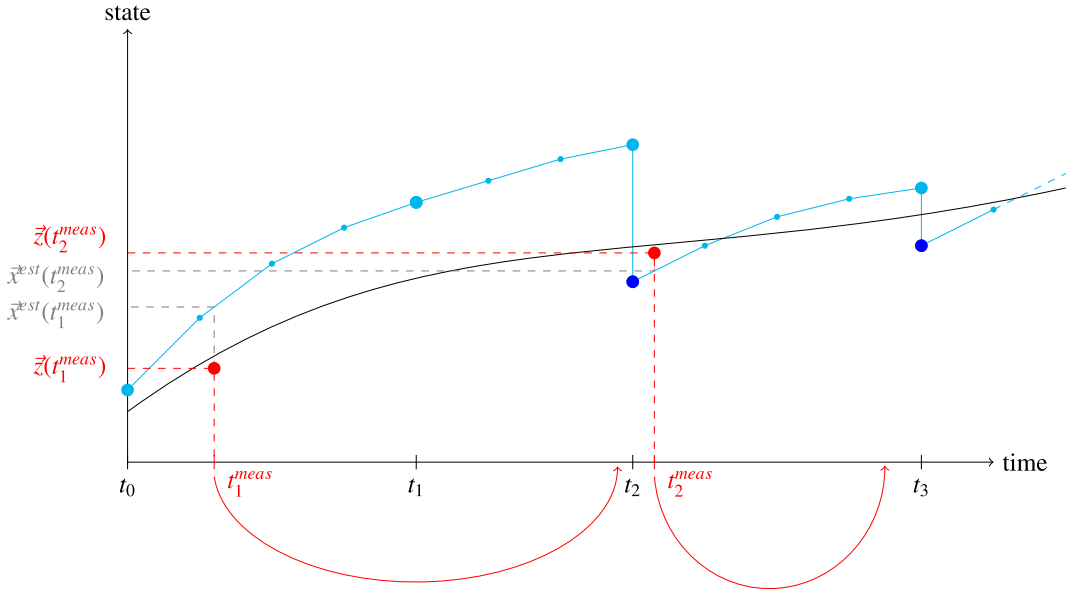


Fig. 2. Example A: Two measurements (red, measurements of type case 1) of the state (black) improve the filter propagation value (light blue) and lead to filter corrections (blue). The second measurement is started after the first measurement is used by the filter.

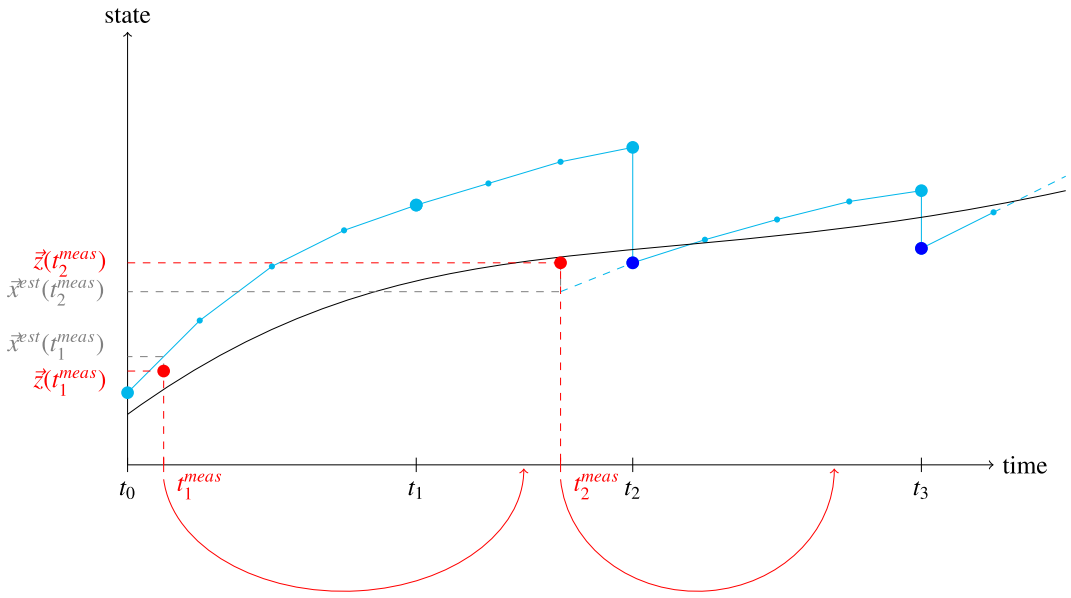


Fig. 3. Example B: Two measurements (red, first measurement of type case 1, second measurement of type case 2) of the state (black) improve the filter propagation value (light blue) and lead to filter corrections (blue). The second measurement is started before the first measurement is used by the filter.

Case 2. Fig. 3 shows a second example. The measurement at time t_1^{meas} is similar as in the previous example. At time t_2 it can be used in the filter correction step. Compared to the example of case 1 the timing is a bit different. In the second case presented in Fig. 3 the second measurement at time $t_2^{meas} \in (t_1, t_2)$ was started before the first measurement was used by the filter correction step at time t_2. Situations like in case 2 can occur in practice if measurement system and filter system are not time-synchronized, for example if they are executed in different independent tasks or on different computing nodes.

An estimate $\vec{x}^{est}\left(t_2^{meas}\right)$ is now not computed by propagating from $\vec{x}_1^{est}$. At time t_3 when the measurement can be used, all most recent knowledge about the filter state should be used including the improvement of the estimate after the first measurement has been included. The estimate $\vec{x}_2^{est}$, obtained after correction, is assumed to be more accurate compared to the predicted value $\vec{x}_2^{pre}$. This is why we use the value $\vec{x}_2^{est}$ and propagate backwards from t_2 to t_2^{meas}.

So for case 2 the general rule is: If a filter correction has been performed at a time $t_s \geqslant t^{meas}$, for $s < k+1$, the estimate $\vec{x}^{est}(t^{meas})$ is computed via backwards propagation from t_s to t^{meas}.

2.2. Sensor fusion

A Kalman filter like estimator is well suited for sensor fusion. Measurements of different sensors can be fused in the filter correction step.

The method described in this paper takes each measurement $\vec{z}(t^{meas})$ as soon as it is provided to the filter and uses it in the next correction step. The source of the measurement is not important. Let $M \in \mathbb{N}$ be the number of sensors. Each sensor generates measurements that need not be time-synchronized. When the filter is executed, there can be $0, 1, \ldots, M$ available measurements. In case of zero measurements, the filter propagates only. In case of only one measurement at once, the filter correction is done as described in sub-Section 2.1.4. For 2 or more measurements the correction method has to be slightly adapted. We note, that the number of available measurements differ in general from one filter time step to the next.

To handle M measurements, we adapt the size of the measurement vector, of the measurement matrix and of the measurement covariance matrix. At time t_{k+1} we consider up to M single measurement vectors $\vec{z}^{(1)}\left(t_{i_1}^{meas}\right), \ldots, \vec{z}^{(M)}\left(t_{i_M}^{meas}\right)$. Each vector is of dimension m. In general, the measurement times $t_{i_1}^{meas}, \ldots, t_{i_M}^{meas} \leqslant t_{k+1}$ need not be equal.

Let $H^{(1)}\left(t_{i_1}^{meas}\right), \ldots H^{(M)}\left(t_{i_M}^{meas}\right) \in \mathbb{R}^{m \times n}$ be the single measurement matrices obtained by the partial derivatives of the measurement model for each sensor, evaluated at $\vec{x}^{est}\left(t_{i_j}^{meas}\right)$ and at time $t_{i_j}^{meas}$, for $j = 1, \ldots, M$.

We define the so-called main measurement vector

$$\vec{z}_{k+1} = \begin{pmatrix} \vec{z}^{(1)}\left(t_{i_1}^{meas}\right) \\ \vdots \\ \vec{z}^{(M)}\left(t_{i_M}^{meas}\right) \end{pmatrix} \in \mathbb{R}^{M \cdot m}, \tag{6}$$

and the main measurement matrix

$$H_{k+1} = \begin{pmatrix} H^{(1)}\left(t_{i_1}^{meas}\right) \\ \vdots \\ H^{(M)}\left(t_{i_M}^{meas}\right) \end{pmatrix} \in \mathbb{R}^{(M\cdot m)\times n}. \tag{7}$$

For each sensor a measurement covariance matrix $R^{(1)}\left(t_{i_1}^{meas}\right), \ldots, R^{(M)}\left(t_{i_M}^{meas}\right) \in \mathbb{R}^{m\times m}$ is given. The main measurement covariance matrix is defined as the following block diagonal matrix:

$$R_{k+1} = \begin{pmatrix} R^{(1)}\left(t_{i_1}^{meas}\right) & 0 & \ldots & 0 \\ 0 & R^{(2)}\left(t_{i_2}^{meas}\right) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & R^{(M)}\left(t_{i_M}^{meas}\right) \end{pmatrix} \in \mathbb{R}^{(M\cdot m)\times(M\cdot m)}. \tag{8}$$

The gain matrix K_{k+1} for filter step t_{k+1} is computed as in (2c), but is of dimension $n \times (M \cdot m)$.

It is now simple to handle the case when not all M measurements are available at the same time. If the measurement of sensor $j, j \in \{1, \ldots, M\}$ is not available at t_{k+1}, the measurement matrix $H^{(j)}$ is set to zero. In this case the corresponding m lines of the j-th block in the main measurement matrix H_{k+1} are zero.

Up to now, we have assumed that all sensors produce the same size of measurement vector, i.e. a vector of dimension $m \in \mathbb{N}$. However, the method is not restricted to this case. For example, it can happen that one sensor measures two physical quantities, and another sensor measures only one quantity. Or one sensor measures a 3D position, whereas another sensor measures the distance only. In the general case, we can assume $\vec{z}^{(1)}\left(t_{i_1}^{meas}\right) \in \mathbb{R}^{m_1}, \ldots, \vec{z}^{(M)}\left(t_{i_M}^{meas}\right) \in \mathbb{R}^{m_M}$. The individual blocks in the main measurement vector $\vec{z}$, in the main measurement matrix H and in the filter covariance matrix are not of the same size. But all calculations can be done the same way as presented above. For each sensor $j, H^{(j)}$ has to be computed by taking the partial derivatives of the measurement model with respect to the state. The single sensors can have different measurement models denoted by the superscript (j).

2.3. Application to rendezvous

In this section we describe how the filter method can be applied to navigation problems arising during spacecraft rendezvous. As reference mission, we consider a debris removal mission where an active spacecraft called chaser approaches a completely non-cooperative, passive spacecraft, called target.

2.3.1. Coordinate frames

The Local Vertical Local Horizontal (LVLH) system, also called spacecraft local orbital frame (Fehse, 2003), is a well suited coordinate system for relative GNC during rendezvous or proximity operations. In our application, close range rendezvous with a non-cooperative target, both our guidance subsystem and our controller use the LVLH system.

R-bar (z-axis of the LVLH system) points radial from the spacecraft to the Earth is thus defined by the absolute position of the target spacecraft. H-bar (y-axis) is obtained by the cross-product of the target's absolute position vector and velocity vector and V-bar (x-axis) completes the right hand system. Since the target is fully non-cooperative, the axes V-bar, H-bar and R-bar are a priori not known and have to be determined. The LVLH frame is not an inertial frame, it changes over time with the motion of the target in its orbit around the Earth, so the estimation has to be done at each execution of the GNC loop.

To get the origin and the axes of the LVLH system, we estimate therefore the position and velocity of the target in an inertial frame, the Earth Centered Inertial (ECI) frame.

In addition to LVLH and ECI, the body frames of chaser and target are involved. The absolute position and orientation of the spacecrafts determine the transformation between ECI and body frame and vice versa.

The relative position and attitude between chaser and target are measured with optical sensors, for example with cameras (Fehse, 2003; Persson et al., 2006; Benninghoff et al., 2014). The measurements are the pose, i.e. position and attitude, of the target in the sensor frame, see Section 2.3.2 below. With the knowledge of the fixed transformation from sensor system to chaser system (determined by a hand-eye calibration) the measurements of the relative position and attitude can be transformed from the sensor coordinate frame to the chaser body frame. The chaser body frame is very useful to compare measurements of different sensors, or to compare filter estimation with measurements by some post-processing (see Section 4).

2.3.2. State and measurement vectors

During rendezvous to a completely non-cooperative passive target spacecraft by a chaser the state of the target should be estimated.

As discussed in Section 2.3.1 we have to estimate the position and the velocity of the target in ECI to get the origin and orientation of the LVLH system which is used by other sub-systems like guidance in the GNC system. Furthermore, the orientation and the attitude rate of the target are important during close range rendezvous, for example when a fly-around should be done to approach the target from a certain geometrical direction with respect to its body frame. Attitude and attitude rate are also an important input for capture of a spinning or tumbling target after a successful rendezvous.

In summary, in our application, the state consists of the absolute position of the target, its velocity, its orientation and its attitude rate.

The state vector of the target can be written as

$$\vec{x} = \begin{pmatrix} \vec{p}_{tar}^{ECI} \\ \vec{v}_{tar}^{ECI} \\ \vec{q}_{tar}^{ECI} \\ \vec{\omega}_{tar}^{ECI} \end{pmatrix} \in \mathbb{R}^{13}, \tag{9}$$

where $\vec{p}_{tar}^{ECI} \in \mathbb{R}^3$ denotes the position vector, $\vec{v}_{tar}^{ECI} \in \mathbb{R}^3$ the velocity vector, $\vec{q}_{tar}^{ECI} \in \mathbb{R}^4$ is the attitude expressed as quaternion, and $\vec{\omega}_{tar}^{ECI} \in \mathbb{R}^3$ is the attitude rate.

The sensors determine the pose (i.e. position and attitude) of the target in sensor frame. The measurement vector can be written as

$$\vec{z} = \begin{pmatrix} \vec{p}_{tar}^{sen} \\ \vec{q}_{tar}^{sen} \end{pmatrix} \in \mathbb{R}^7. \tag{10}$$

2.3.3. Orbit and attitude dynamics and kinematics

For the filter propagation step, system dynamic equations for the orbit and attitude of the target in ECI coordinates have to be solved. To simplify the notation, we use $\vec{p} = \vec{p}_{tar}^{ECI}$ and similarly $\vec{v}, \vec{q}$ and $\vec{w}$ in the following. For our tests, we model the orbit using the two body problem (Newton's law of gravity):

$$\frac{d^2}{dt^2}\vec{p} = -\mu \frac{\vec{p}}{\|\vec{p}\|^3} + \frac{\vec{F}}{m_{tar}}, \tag{11}$$

where $\mu = G \cdot M_{Earth} = 398600.4415 \cdot 10^9 \mathrm{m}^3/\mathrm{s}^2$ is the Gravitational constant multiplied with the Earth's mass, $\|\vec{p}\|$ denotes the Euclidean norm of the vector $\vec{p}, m_{tar}$ is the mass of the target and $\vec{F} \in \mathbb{R}^3$ is an input or disturbance vector. For the rendezvous problem, $\vec{F}$ is often unknown. This is why we often model this uncertainty with additive noise, see Eq. (1a).

The corresponding discrete linear equations for the position and the velocity can be written as:

$$\vec{p}_{k+1} = \vec{p}_k + (t_{k+1} - t_k)\vec{v}_k + \vec{v}_k^{(p)}, \tag{11a}$$

$$\vec{v}_{k+1} = \vec{v}_k - (t_{k+1} - t_k)\mu \frac{\vec{p}_k}{\|\vec{p}_k\|^3} + \vec{v}_k^{(v)}, \tag{11b}$$

with system noise $\vec{v}_k^{(p)}$ for the position and $\vec{v}_k^{(v)}$ for the velocity. As before, the index k denotes that the value is evaluated at t_k.

From Eq. (11) and the corresponding law for the chaser, equations of the relative motion can be derived. For circular orbits, these are the well-known Hill equations:

$$\frac{d^2}{dt^2}x - 2\omega\frac{d}{dt}z = \frac{F_x}{m_{cha}} \tag{12a}$$

$$\frac{d^2}{dt^2}y + \omega^2\frac{d}{dt}y = \frac{F_y}{m_{cha}} \tag{12b}$$

$$\frac{d^2}{dt^2}z + 2\omega\frac{d}{dt}x - 3\omega^2\frac{d}{dt}z = \frac{F_z}{m_{cha}} \tag{12c}$$

The derivation of the Hill equations can be found in Fehse (2003) (Appendix A). The main idea is a linearization by a Taylor series expansion. In the Hill equations, ω is the angular frequency of the circular target orbit. $\vec{p}_{cha}^{LVLH} = (x, y, z)^T$ is the position vector of the chaser in the local orbital frame of the target, the LVLH-frame with the axis V-bar (x-axis), H-bar (y-axis) and R-bar (z-axis).

Knowing already V-bar, H-bar and R-bar as well as the angular frequency ω, the Hill equations are often used in rendezvous applications. Its closed form solution is given by the so-called Clohessy-Wiltshire equations, see Fehse (2003).

The attitude dynamics are modeled via the quaternion differential equation and the Euler equation. The quaternion differential equation is given by

$$\frac{d}{dt}\vec{q} = \frac{1}{2}\Omega\vec{q}, \tag{13}$$

where $\Omega \in \mathbb{R}^{4\times4}$ is defined by

$$Q = \begin{pmatrix} 0 & -\omega_x & -\omega_y & \omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{pmatrix}. \tag{14}$$

The Euler equation is

$$I\frac{d}{dt}\vec{\omega} = \vec{T} - \vec{\omega} \times (I\vec{\omega}), \tag{15}$$

where $I \in \mathbb{R}^{3\times3}$ is the matrix containing the moments of inertia and $\vec{T} \in \mathbb{R}^3$ is the torque vector acting on the target. Similarly, as for the force, the torque vector is typically unknown in rendezvous applications with passive targets, so we can set it to zero and add noise vectors.

The corresponding discrete linear equations for attitude and attitude rate can be written as:

$$\vec{q}_{k+1} = \vec{q}_k + (t_{k+1} - t_k)\frac{1}{2}\Omega_k\vec{q}_k + \vec{v}_k^{(q)}, \tag{16a}$$

$$\vec{\omega}_{k+1} = \vec{\omega}_k - (t_{k+1} - t_k)I^{-1}(\vec{\omega}_k \times (I\vec{\omega}_k)) + \vec{v}_k^{(\omega)}, \tag{16b}$$

with system noise $\vec{v}_k^{(q)}$ for the attitude and $\vec{v}_k^{(\omega)}$ for the attitude rate. I^{-1} denotes the inverse of I, i.e. here a linear equation has to be solved (via QR-method or similar methods).

In Section 2.1.3 we mentioned that the propagation should be split in sufficiently small time steps. For the linear propagation of the orbit (11) we use a time step of 0.004 s, and for the linear propagation of the attitude and attitude rate (16) we use a time step 0.1 s.

The selection of the time step sizes were done based on the used test environment, see Section 3. The command rate of 250 Hz of our test facility was used for the propagation of the orbit. For the attitude we used a multiple of the sample time. We did not investigate on optimal time step sizes or the influence of the time step sizes on the accuracy of the propagation, which is beyond the scope of this paper. In general, however, the reader should be aware that linear propagation has to be handled with care, as discussed in textbooks on numerical mathematics such as the book of Stoer and Bulirsch (2002).

2.4. Implementation details

The method for a navigation filter with delayed measurements and sensor fusion can be implemented in various ways and programming languages. We have implemented it as a C++-class which enables us to use it for our C++-based guidance, navigation and control library.

As mentioned in Section 2.1.4, a number of previous state estimates have to be stored to handle measurements with delay. This can be done by using a ring buffer of a fixed, sufficiently large size. In our application, a ring buffer is used which can store 200 elements. A fixed-size ring buffer allows for high performance since its size needs not to be changed. Using a so-called *push-front* routine, a new state is included in the ring buffer at the front of the buffer. If the buffer is already full, one element at the back of the buffer is removed (= the oldest one).

Since the time of measurements and states is important to handle the delay, the state and the measurements are also implemented as C++-classes, storing the time and the state/measurement components. Time and other data can be accessed via "get" and "set" functions. Using the time information, the buffer elements can be found by comparing the times of the stored states with the time of the delayed measurements.

Different combinations of available, new measurements can occur during sensor data fusion, see Section 2.2. All sensors can provide new measurements, or only a subset. Dependent on the application, it may also be interesting to avoid the use of sensor measurements for some duration. For example during the initialization phase of a sensor.

To handle these requirements, we use two bit masks. One bit mask is called *active sensors* that contains the information which sensors are active. The second is called *used sensors* that contains the information which sensor data is actually used in the current filter correction step. The two bit masks are stored as unsigned integer with 8 bits, since 8 different sensors are sufficient.

Since the guidance, navigation and control system should be used in the future as on-board software on a chaser satellite, the bit mask *active sensors* is configured via tele-commands. From an on-ground console, the operator can first start camera image processing and then switch it to active by two separate tele-commands. When the second tele-command is sent, the bit in the *active sensors* bitmask corresponding to the camera, is set to 1. By tele-commands each single sensor can be activated and the bit mask on-board informs the filter which sensor has been activated.

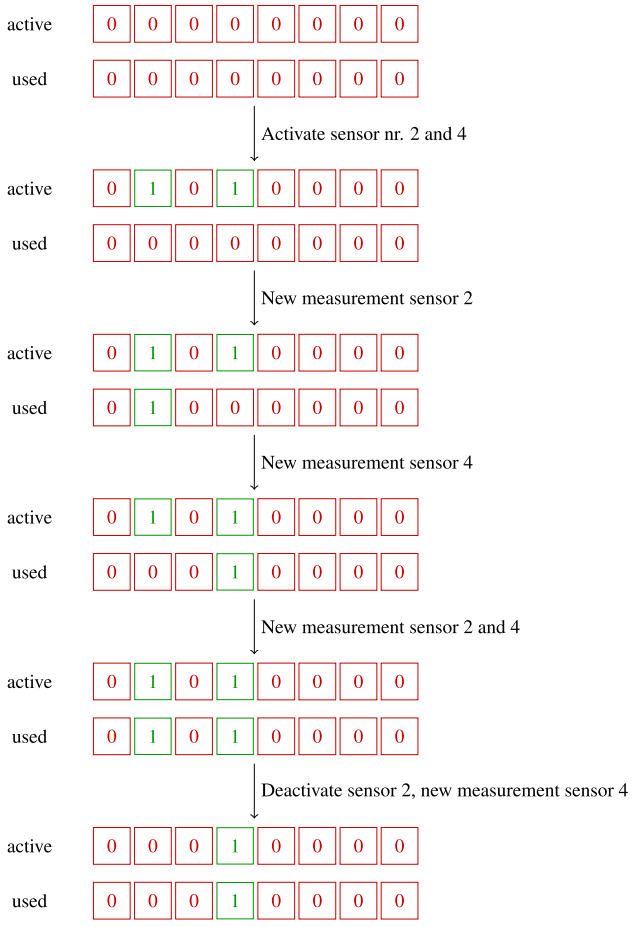


Fig. 4. Example showing how bit masks for active sensors and used sensors are set dependent on activation of sensors and new measurements.

The *used sensors* bit mask can be understood as a subset of the activated sensors, those sensors which provide a new measurement which has not yet been used by the filter in a previous step. This is needed, as explained above, because the filter can be executed more often compared to the generation of new measurements.

The *used sensors* bit mask thus typically changes every filter execution step, whereas the *active sensors* bit mask only changes when a tele-command is sent to activate or deactivate a sensor for the filter.

Fig. 4 presents an example. At the beginning no sensor is active. Then two sensors, number 2 and 4, are activated and the corresponding bits are changed to 1. However no measurement is ready to be used by the filter. Since the bit mask of used sensors is zero, the filter propagates only. In the next step, there is a ready measurement of sensor 2. The corresponding bit mask is set to 1. The filter uses the measurement in its correction step. In the next step, there is a new measurement of sensor 4, but no new measurement of sensor 2. The filter thus uses the measurement of sensor 4. In the next step, both sensors provide a new

measurement. So the filter uses both measurement. In the last time step, sensor 2 is deactivated and sensor 4 generates a new measurement and this measurement can be used by the filter.

3. Test environment

As mentioned in Section 2.3 we apply the navigation filter with delay and sensor fusion to a rendezvous application. We test the filter and the entire guidance, navigation and control system within a HiL, end-to-end simulation framework. This section describes the test-environment.

3.1. Simulation facility EPOS 2.0

The space segment of the rendezvous phase of an on-orbit servicing mission or other spacecraft rendezvous mission is simulated at European Proximity Operations Simulator (EPOS), a large-scale research facility for HiL-simulation of rendezvous, inspection and close proximity scenarios (Benninghoff et al., 2017; Rems et al., 2021).

EPOS is a robotic testbed with two robots, see Fig. 5. Each robot has six degrees of freedom. One robot is mounted on a linear slide of 25 m length, see also Fig. 6. Thus, the final 25 meters of the close range rendezvous phase can be simulated with 1:1 models (and even larger distances with scaled models). One of the two EPOS robots simulates the motion of the chaser spacecraft during the final rendezvous phase, the other robot simulates the motion of the target spacecraft.

EPOS can be used for test campaigns (Mühlbauer et al., 2013; Burri et al., 2021; Rems et al., 2021) and research projects (Benninghoff et al., 2018; Klionovska and Burri, 2021). A typical setup for rendezvous can be seen in Fig. 5. One robot carries an adapter board where several sensors are mounted. The second one carries a mockup of a target satellite. EPOS can be used for *open loop tests*, where the robots follow a predefined trajectory and sensor data is collected for off-line post-processing and comparison with the EPOS ground-truth. However, we mainly

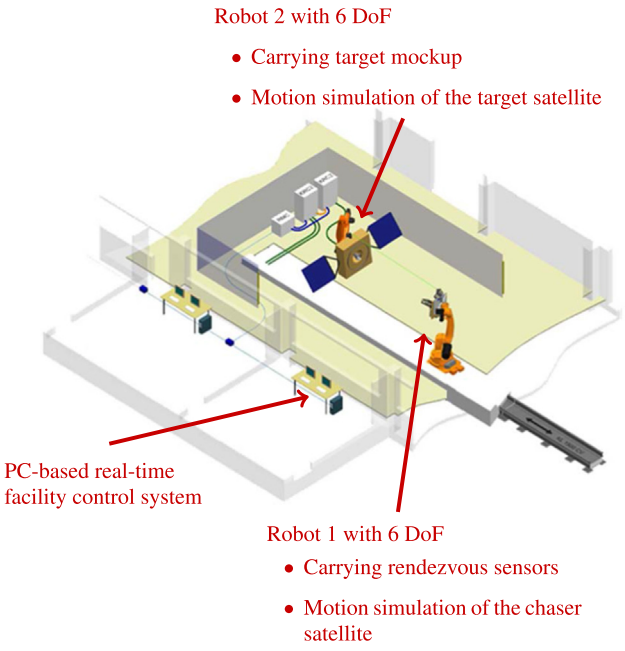


Fig. 6. Overview on the EPOS laboratory, the robots and the control system.

use EPOS in our research projects for *closed loop tests* (Benninghoff et al., 2014; Benninghoff et al., 2018): Sensor data like camera images are processed in real-time, the result is used by a navigation filter, the guidance delivers a reference trajectory and the controller, which compares estimated values with guidance values, computes commands for the actuator system. The actuators and the spacecrafts' orbit and attitude dynamics are simulated in software (satellite simulator) and the resulting motion is commanded to the facility. The facility has a command rate of 250 Hz (Benninghoff et al., 2017).

3.2. Rendezvous sensors

At EPOS, different optical sensors can be integrated and tested. We use two different optical sensors for demonstra-

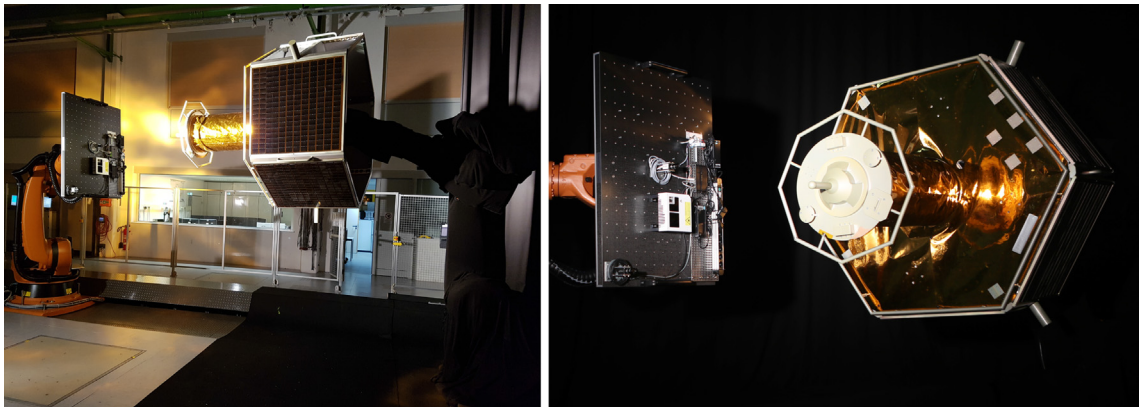


Fig. 5. The robots at the EPOS facility during a typical rendezvous HiL simulation. One robot carries rendezvous sensors, the second robot carries a mockup of a target spacecraft.

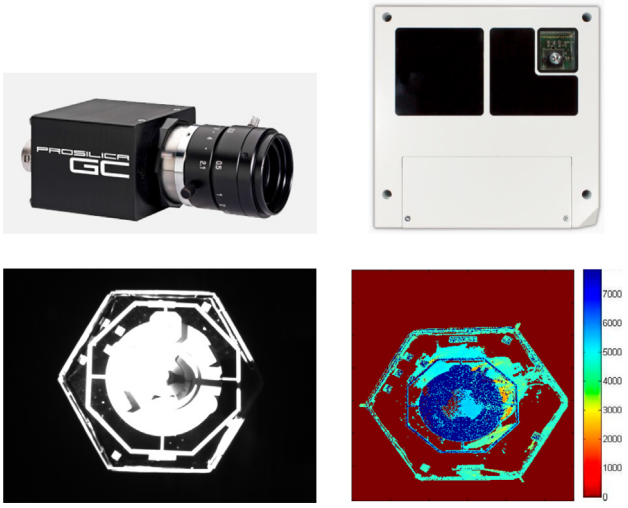


Fig. 7. Sensors used for sensor fusion demonstration: Left: 2D monocular camera (Prosilica GC-655) and example gray-scaled image. Right: photonic mixer device camera (Bluetechnix, DLR-Argos 3D-P320 camera prototype) and example depth image.

Table 1
Characteristics of the rendezvous sensors used for the filter tests

Sensor name	Field of view [deg]	Resolution [pixels]
2D monocular camera (Prosilica GC-655)	53 × 41	640 × 480
PMD (Bluetechnix, DLR-Argos 3D-P320)	28.91 × 23.45	352 × 287

tion of the navigation filter with delay and sensor fusion: a 2D monocular camera and a Photonic Mixer Device (PMD) sensor, see Fig. 7.

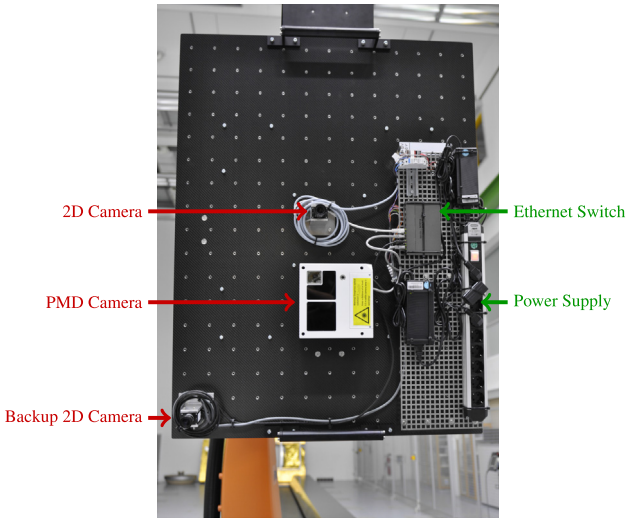


Fig. 8. Sensor adapter plate mounted at one of the EPOS robots with 2D mono camera and PMD camera.

The 2D camera provides an intensity image with a resolution of 640 x 480 pixels, see Table 1. The PMD camera is an active sensor which determines the distance by considering the phase-shift of emitted and reflected signals (Klionovska et al., 2018). It generates images of intensity value and measured distance in each frame with a resolution of 352 x 287 pixels each. Fig. 8 shows one robot of the EPOS facility with a sensor adapter board.

The two sensors are used at EPOS as rendezvous sensors for approaches with a 1:1 model of a typical target satellite (see example images in Fig. 7). The PMD camera is limited for ranges between 5 m and 8 m only. The upper limit of 8 m is due to the physical measurement principle which is based on the phase shift between sent and received signal. Up to 8 m an ambiguity free measurement can be performed in a stable way. The lower limit of 5 m is due to the field of view of the PMD camera. In a typical test case we perform the approach starting with the 2D camera as primary navigation sensor. At 8 m, the PMD camera is also used by the filter. The second part of the approach is thus done with sensor fusion.

We can tune the filter slightly by setting the values of the noise covariance matrices, see Section 2.2. The 2D camera has a higher resolution compared to the PMD camera. This is why lateral motions can be measured with higher accuracy by the 2D camera compared to the PMD camera. However, since the PMD sensor directly delivers physical distance measurements, the accuracy of the distance component of the relative position is more accurate using the PMD camera compared to the 2D monocular camera.

3.3. GNC system

The navigation filter is part of a GNC system for rendezvous with the components guidance, navigation and control.

The navigation component contains the physical rendezvous sensors, pose estimation software for each sensor and the navigation filter software. The pose estimation systems generate measurements of the relative pose of the target with respect to the chaser. The filter processes these measurements together with the current absolute position and orientation of the chaser to generate an estimation of the state of the target in ECI coordinates.

The guidance component generates a smooth trajectory in the LVLH system which the chaser should follow. The LVLH system is defined by the filter output (position and velocity of the target in ECI determine the axes of the LVLH system, see Section 2.3.1.) The guidance trajectory consists of several sub-trajectories whose parameters can be commanded from ground. Via tele-command, parameters like desired distance to the target, desired approach velocity, or desired elevation or azimuth angles for a fly-around can be set. Also the attitude can be changed by commanding desired roll, pitch or yaw angles. The guidance trajectory with values for each time step is then computed autonomously on-board using the previously

commanded parameters. At the end of each sub-trajectory the guidance enters a so-called hold position mode automatically and waits for new commands.

The control component finally computes the necessary forces and torques for the chaser satellite by comparing its actual values with the guidance values. The controller is a linear-quadratic regular with an integration term.

3.4. On-board computer

In this paper, we also present results where the on-board GNC system is executed on a representable On-Board Computer (OBC) hardware. For our tests we use the Computing for Space Avionics (ScOSA) OBC (Treudler et al., 2018; Lund et al., 2022). The ScOSA OBC combines highly reliable space qualified hardware with highly performant commercial off-the-shelf (COTS) components called High Performance Nodes (HPN) based on a Xilinx Z7020 dual-core ARM Cortex-A9. The nodes are connected via 100 Mbit/s Ethernet for the laboratory configuration (space wire is used for the space configuration). Lund et al. (2022) present a ScOSA tasking framework such that applications can be implemented using different tasks interconnected by channels. This allows to asynchronously execute the computational expensive tasks image processing and pose estimation on a different node than the rest of the GNC System.

We chose this platform to present and discuss results with representative computational times and thus for realistic sensor delays produced by the image processing and pose estimation algorithms.

4. Results

In this section the performance of the navigation filter is presented for different cases: First, we use only one sensor and test the filter at two different hold points (test case 1 and 2). Next, still using one sensor, we test the system during an entire approach (test case 3). We repeat the approach with sensor fusion, i.e. we use both a monocular camera and a PMD camera (case 4) and finally, we investigate the effect of realistic delays produced by executing the GNC system on the ScOSA OBC (case 5). All cases are simulated in closed-loop; the filter results are compared with the guidance trajectory in a controller to generate actuator commands for the simulated chaser, in turn changing the chaser's motion in the EPOS laboratory (cf. Section 3).

Table 2
Overview on test conditions for case 1 (hold point at 15 m), coordinates of the target in chaser body frame (Euler angle convention 321).

	Initial	During test case
p_x [m]	15	approx. constant
p_y [m]	0	approx. constant
p_z [m]	0	approx. constant
e_A [deg]	−90	spinning 1 deg/s
e_B [deg]	0	approx. constant
e_C [deg]	180	approx. constant

4.1. Case 1: single sensor at hold point at 15 m distance

In the first test scenario the navigation filter receives measurements of a single sensor. For this test we use the Prosilica camera, a 2D monocular camera, at EPOS, see Fig. 7 (left). We analyze results obtained at a fixed hold point at about 15 m distance to the target. Hold point means that the guidance value is constant.

Table 2 shows the test conditions. The chaser body frame (see Section 2.3.1) is used for the table due to presentation reasons and in accordance with most of the following figures. The Euler angle convention is 321, i.e. starting from the chaser frame, the orientation of the target frame is obtained by a rotation with angle e_C around the chaser z-axis, a rotation with angle e_B around the resulting y-axis, followed by a rotation with angle e_A around the resulting x-axis. The initial orientation of the target frame with respect to the chaser frame is visualized graphically in Fig. 9.

As described in Section 2.3.1, the filter estimates the target's state in the ECI coordinate frame. Fig. 10 shows the filter estimates in ECI: the estimation of the position, the velocity, the attitude quaternion and the attitude rate of the target.

We discussed in Section 1.4 that orbit dynamics leads to high velocities of several kilometer's per seconds (here: at the start point about 7 km/s in negative z-direction, +/- 2 km/s in x- and y-direction). This is why noise or oscillations of the filter cannot be observed when looking at the resulting position and velocity plots in ECI coordinates. Considering the attitude however, some noise and oscillations can be recognized. From the attitude rate it can be seen that the attitude motion in y and z is approximately zero, and in the x-coordinate there is a motion of approximately 1 deg/s (0.017 rad/s), see also Table 2.

For a better comparison with the raw sensor measurements, the chaser body frame can be used to analyze the results. The estimation of the target's pose by the filter in

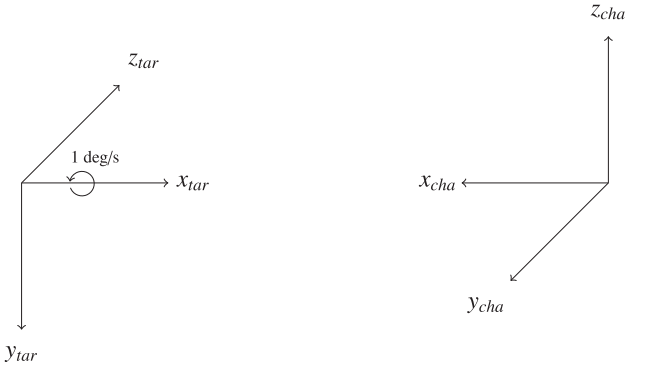


Fig. 9. Drawing and visualization of the initial orientations of target and chaser body frame with respect to each other.

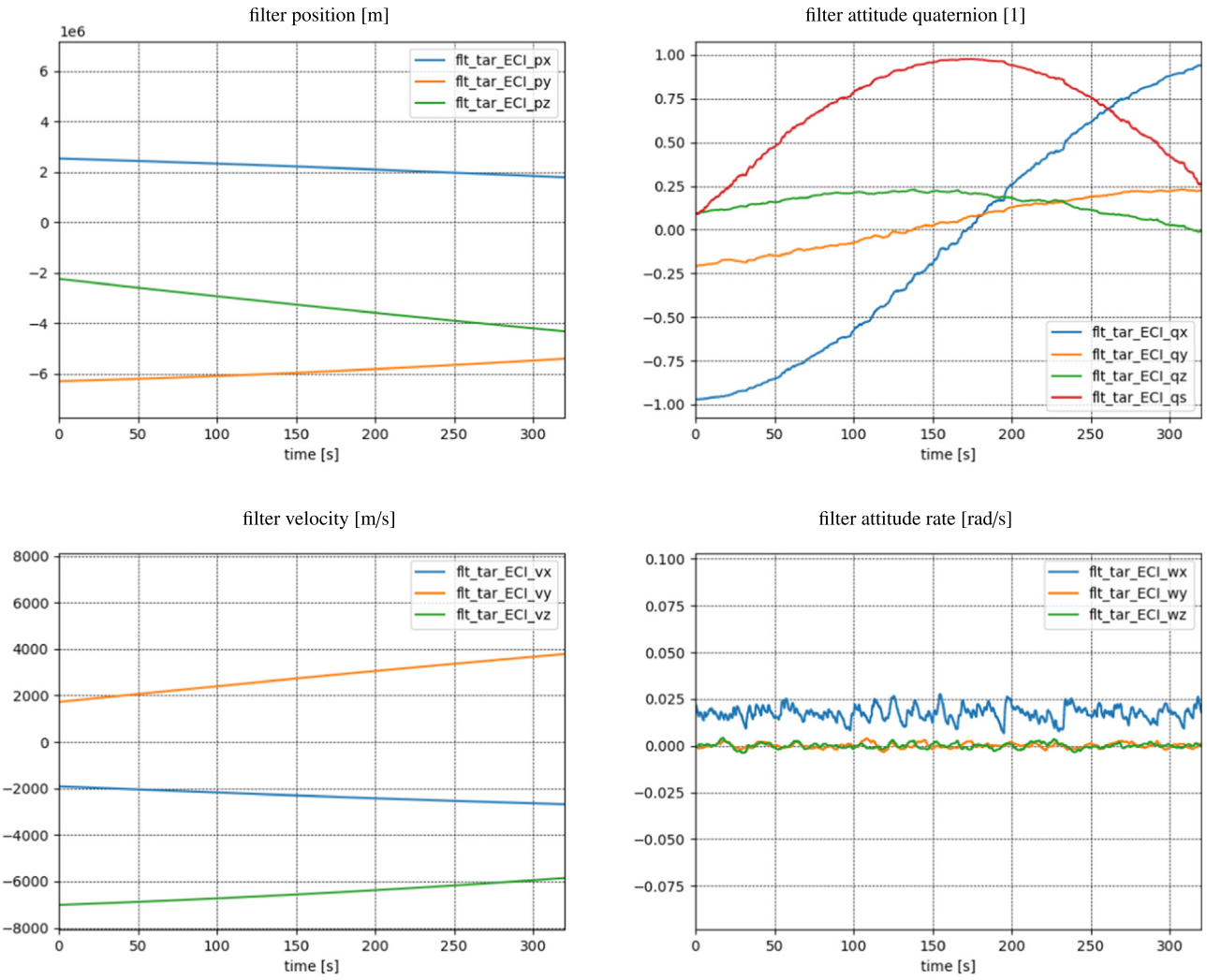


Fig. 10. Results at a hold point at 15 m using a monocular camera as sensor - filter estimates of position, velocity, attitude quaternion and attitude rate in ECI.

ECI can be transformed to an estimation of the target's pose in the chaser body system (denoted with CHA in the plots) because the chaser's pose in ECI is known, see Section 2.3.1. Further, the camera measurement, first obtained in its camera body frame, can be transformed to the chaser body system, because the calibration of the camera is known, i.e. the transformation between camera frame to chaser body frame. Therefore in the following, the results are presented with respect to the chaser body system as common coordinate system. Additionally, for post-processing, we know the ground truth of the EPOS robots. The ground truth values can also be converted to the position and orientation of the target in the chaser frame.

The velocities and attitude rates are estimated by the filter, but not measured by the camera. Only position and attitude are available from ground truth. This is why we convert only the position and attitude of the filter estimates to the chaser body system.

Fig. 11 shows the absolute position of the raw camera measurement in one sub-figure for each component x, y and z and the values of the filter estimation in chaser body frame. For comparison we use the ground-truth of the EPOS robots, that is unknown to the filter.

In our setup the x-axis of the chaser body system points towards the target, see Fig. 9. From camera images the distance is difficult to measure, since it has to be computed by using a model of the target and some triangulation technique, see also Benninghoff et al. (2018). The measurement of the translational components (y and z) is more accurate compared to the distance component (x) which can be observed in Fig. 11. The highest noise and the largest error affects the x-measurement and x-filter estimate.

Specifically at this 15 m hold point, the target is relatively small in the camera image. Due to the finite and discrete resolution of the camera image, the measurement has a significant deviation from the true position. (However, as we will see later, the error improves when we reduce the distance to the target.) Keeping in mind that the filter has no information about the true position of the target but only the measurement and knowledge about the underlying

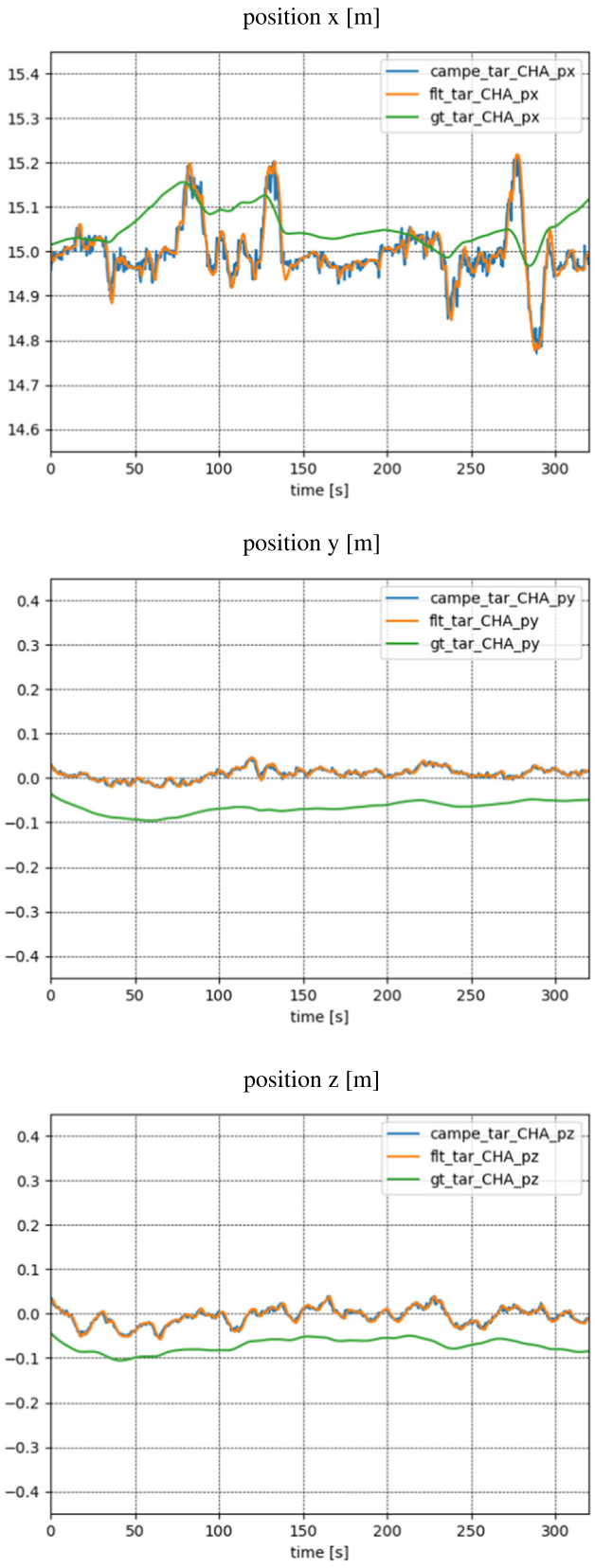


Fig. 11. Performance evaluation at a hold point at 15 m using a monocular camera as sensor - absolute position in chaser body frame (campe = camera pose estimation, flt = filter, gt = ground truth).

dynamics, it is clear that a deviation of the measurement cannot be corrected to 0 by the filter. It has a smoothing effect, can provide estimates with a higher frequency compared to the measurement rate and can estimate also velocities and attitude rates. But it cannot clean a significant deviation.

This case demonstrates that the point at 15 m (x = 15, y = 0, z = 0) can be held in a stable way.

Similarly, we investigate the performance of the attitude estimation. For this we express the attitude in Euler angles, denoted with eA, eB and eC in the following. We use the convention that the rotation from target body frame to chaser body frame can be expressed by three consecutive rotations: first, eA around the x-axis, second, eB around the resulting y-axis and finally, eC around the resulting z-axis.

Fig. 12 shows the absolute values of the attitude measurement, the attitude filter estimates and the ground-truth for comparison. It can well be observed how the target performs a spinning motion (see eA-component). Its spinning axis is aligned with the x-axis. The eB-component is around 0 degrees, and the eC-component is around 180 degrees.

We now use the ground-truth to compute the filter error and compare it with the error of the sensor measurement. Fig. 13 presents the Euclidean error norm between the filter's position and the ground truth and between the sensor measurement and the ground truth.

The attitude error of the filter estimate and the error of the measurement are shown in Fig. 14. The absolute angles between the orientations (between measured orientation and ground truth orientation for the measurement error, and between filter orientation and ground truth orientation for the filter error) are plotted. The maximum error is around 7 degrees. Please note, that this is the estimation of the attitude of the target at the beginning of the close range approach. We will see later, that this value improves when the distance gets smaller. Also note, that this is the estimation error of the orientation of the non-cooperative, spinning target - it is not the pointing error of the chaser. The main contribution to the error is the error of the eA-component. The value of eA changes continuously because of the spinning motion and is a challenge for the pose tracker. The eB- and eC-components are approximately constant and the estimation is more accurate (error smaller than 2 degrees).

As described in Section 2.4 the filter uses two bitmasks *active sensors* and *used sensors*. Fig. 15 shows an illustration of the two bitmasks for the first 4 s of the experiment. We present only a short time span for illustration reasons, otherwise little would be seen in the plot. The filter is executed with approximately 10 Hz, whereas the camera pose estimation runs with approximately 5 Hz. Every second execution of the filter, a new camera measurement is available in this test case.

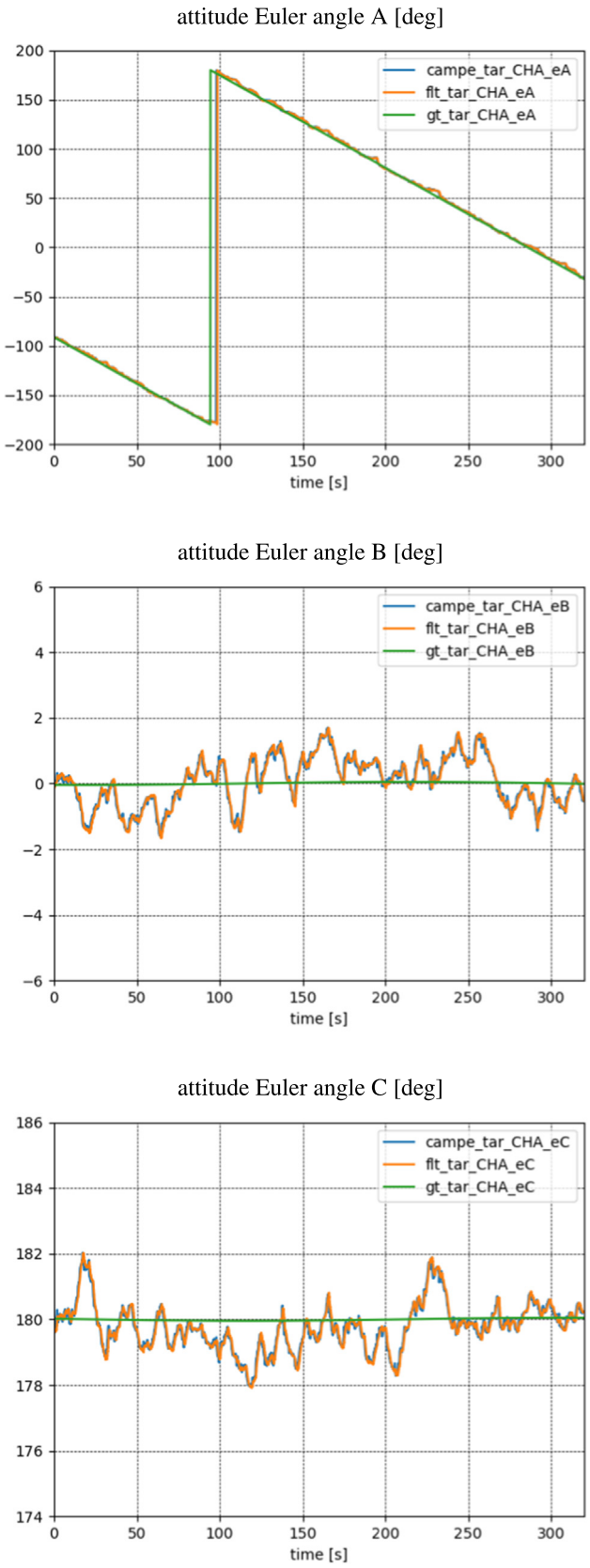


Fig. 12. Performance evaluation at a hold point at 15 m using a monocular camera as sensor - absolute attitude in chaser body frame (Euler angles in degrees, convention 1–2-3) (campe = camera pose estimation, flt = filter, gt = ground truth).

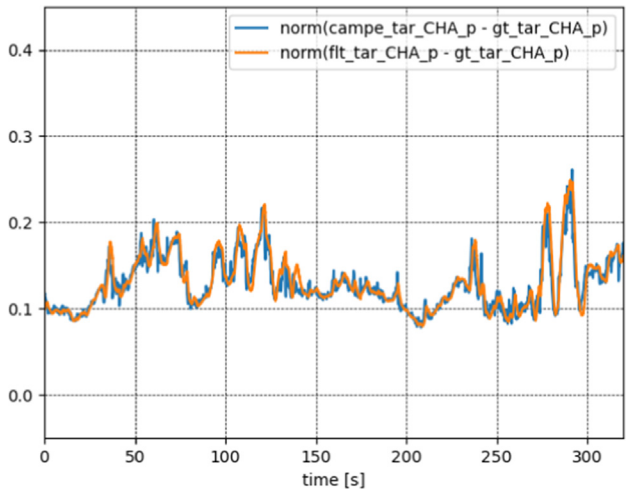


Fig. 13. Performance evaluation at a hold point at 15 m using a monocular camera as sensor - error norm of the position in chaser body frame (campe = camera pose estimation, flt = filter, gt = ground truth).

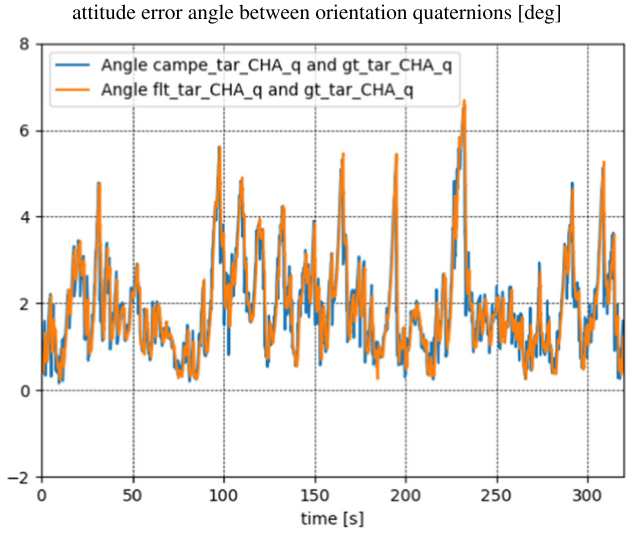


Fig. 14. Performance evaluation at a hold point at 15 m using a monocular camera as sensor - attitude error in chaser body frame (absolute angle between the compared orientations) (campe = camera pose estimation, flt = filter, gt = ground truth).

In Fig. 16 the time delay is presented. It is approximately 0.25 s during this test.

4.2. Case 2: single sensor at hold point at 8 m distance

In the second test case, we analyze the filter performance at a second hold point at 8 m distance to the target. We expect that both the measurement and the filter estimation improve the closer we approach the target.

Table 3 shows the initial conditions and test conditions. The setup is very similar to test case 1, except that the distance is now 8 m and the Euler angle e_A is at 20 deg, when the test is started. (Note, that due to the spinning motion,

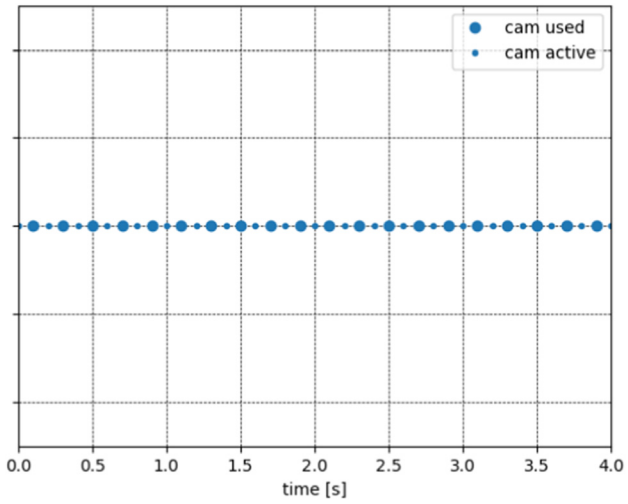


Fig. 15. Hold point at 15 m using a monocular camera as sensor - illustration of active and used sensors at simulation time [0, 4].

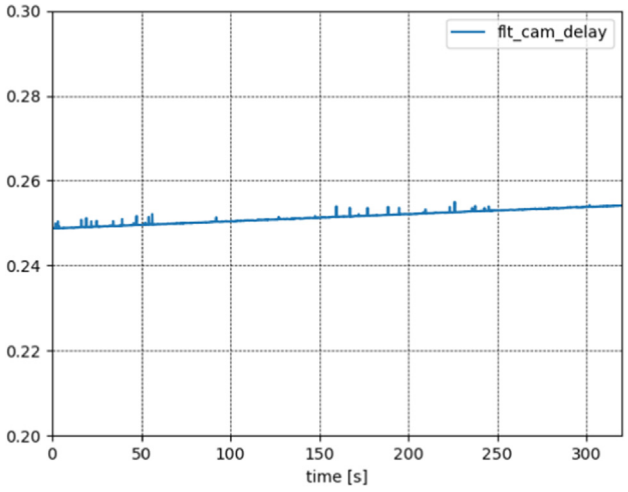


Fig. 16. Hold point at 15 m using a monocular camera as sensor - time delay [s] of the camera measurements.

Table 3
Overview on test conditions for case 2 (hold point at 8 m), coordinates of the target in chaser body frame (Euler angle convention 321).

	Initial	During test case
p_x [m]	8	approx. constant
p_y [m]	0	approx. constant
p_z [m]	0	approx. constant
e_A [deg]	20	spinning 1 deg/s
e_B [deg]	0	approx. constant
e_C [deg]	180	approx. constant

the angle e_A varies. In test case 2, the angle at start of logging was coincidentally at 20 deg.)

Fig. 17 shows the absolute position and Fig. 18 the absolute attitude at the second hold point at 8 m.

The error is presented in Fig. 19 (position error norm). Compared to the previous hold point at 15 m distance to

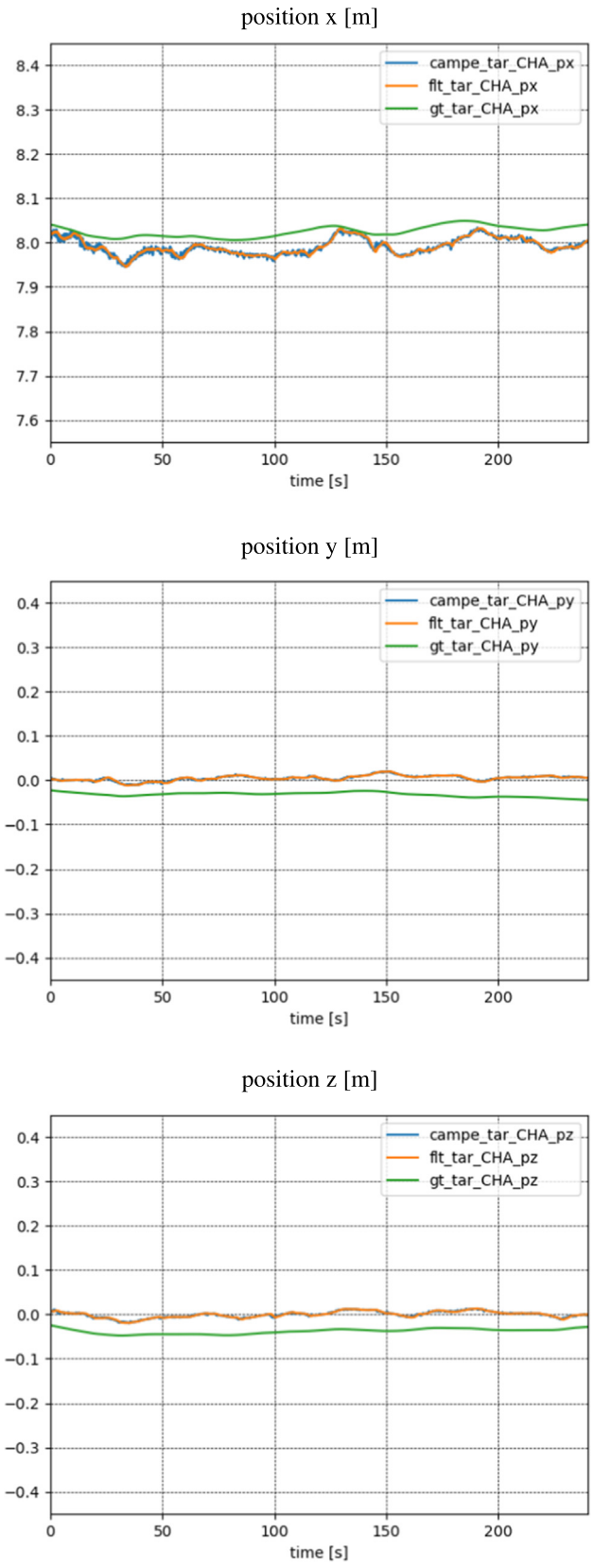


Fig. 17. Performance evaluation at a hold point at 8 m using a monocular camera as sensor - absolute position in chaser body frame (campe = camera pose estimation, flt = filter, gt = ground truth).

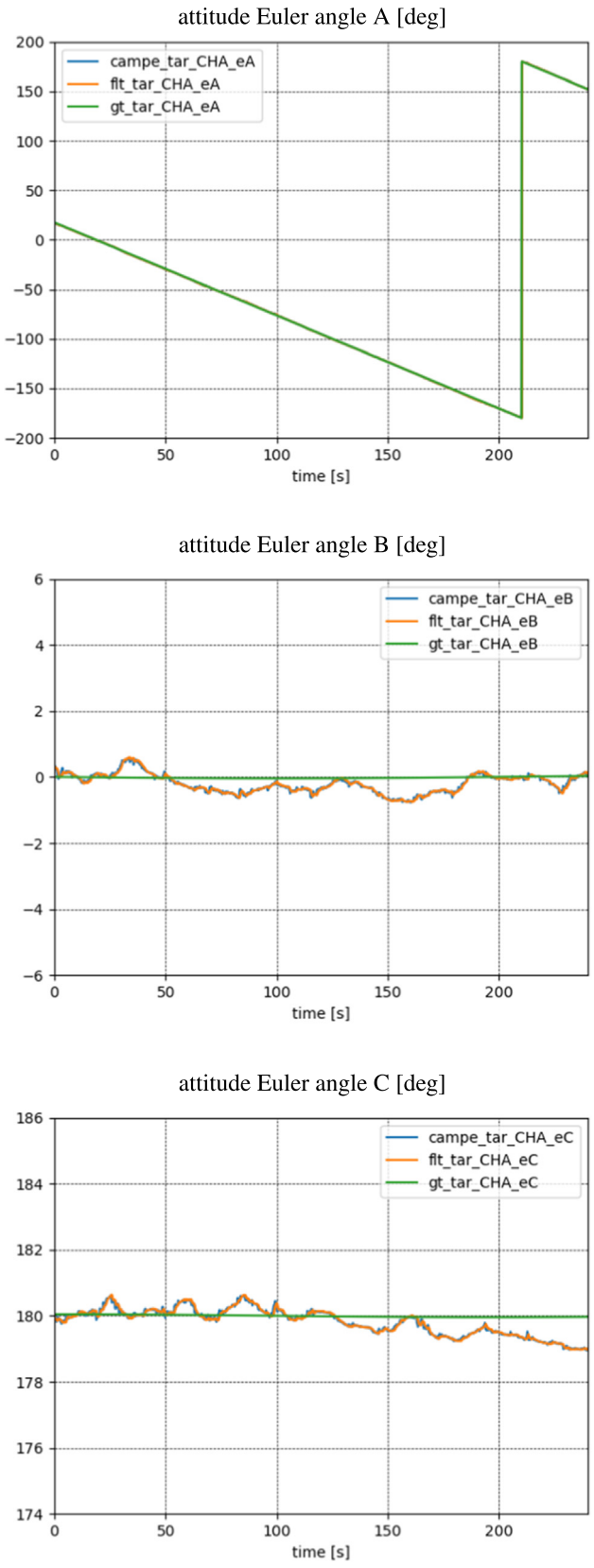


Fig. 18. Performance evaluation at a hold point at 8 m using a monocular camera as sensor - absolute attitude in chaser body frame (Euler angles in degrees, convention 1–2-3) (campe = camera pose estimation, flt = filter, gt = ground truth).

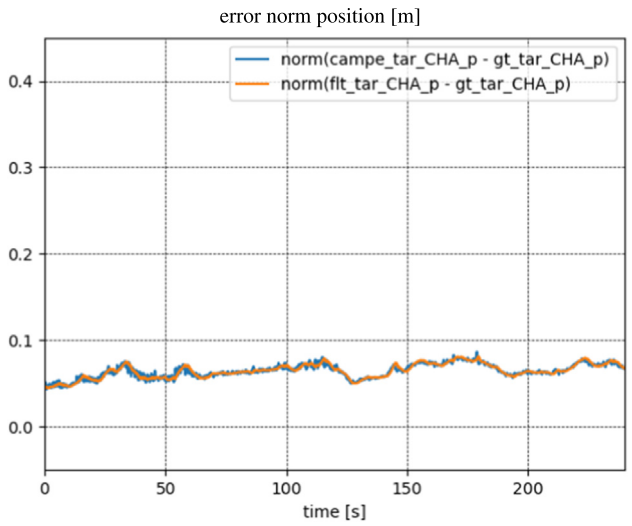


Fig. 19. Performance evaluation at a hold point at 8 m using a monocular camera as sensor - error norm of the position in chaser body frame (campe = camera pose estimation, flt = filter, gt = ground truth).

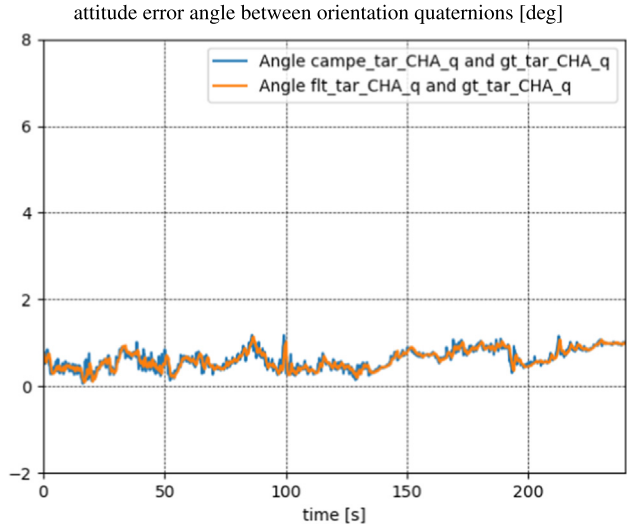


Fig. 20. Performance evaluation at a hold point at 8 m using a monocular camera as sensor - attitude error in chaser body frame (absolute angle between the compared orientations) (campe = camera pose estimation, flt = filter, gt = ground truth).

the target, with position errors up to 10 cm - 20 cm, the measurement of the camera and the filter estimates have both improved. Fig. 20 shows the attitude error (presented as angle between the two compared orientations). The attitude errors at 8 m are also much smaller compared to the attitude errors at 15 m.

4.3. Case 3: single sensor during an approach

We now analyze the performance of the filter during an entire approach from 15 m to 4 m with an intermediate hold point at 8 m. As above for the tests at the hold points,

Table 4
Overview on test conditions for case 3 (approach), coordinates of the target in chaser body frame (Euler angle convention 321).

	Initial	During test case
p_x [m]	15	15 to 8 with 2 cm/s, 8 to 4 with 1 cm/s
p_y [m]	0	approx. constant
p_z [m]	0	approx. constant
e_A [deg]	−110	spinning 1 deg/s
e_B [deg]	0	approx. constant
e_C [deg]	180	approx. constant

we compare the filter estimates for the position and attitude with the camera measurements and with the ground truth of the EPOS robots.

Table 4 shows the initial and test conditions for this test case.

Fig. 21 shows the absolute position and Fig. 22 the absolute attitude of the filter estimate, the camera measurement and the ground-truth in the chaser body frame.

Fig. 23 presents the corresponding position error norm and Fig. 24 the attitude error.

Looking at the x-component of the position (see Fig. 21), different phases of the rendezvous can be observed. The approach starts at 15 m distance to the target. The first phase, until an intermediate hold point at 8 m, is performed by following a guidance trajectory with 2 cm/s approach velocity. At the beginning of the approach, the measurement and the filter estimate in the x-component of the position are stair-shaped. This is due to the fact that a camera has a finite and discrete resolution. When we approach, and the theoretical change of the target's size in the image is less than one pixel, a change in the distance cannot be recognized. This is why we see different constant levels until the change in the image is larger than 1 pixel and the measurement of the distance *jumps* a few centimeters. Especially for distances between 15 m and 11 m this fact can be seen when analyzing Fig. 21, x-component. Similarly to the observations of a stair-like measurement, we can observe oscillations in the error norms, see Fig. 23, especially during the first 200–300 s of the test.

Having reached the intermediate hold point at 8 m, a second approach to the final hold point of this demonstration is done. The final hold point is at 4 m distance to the target. This hold point is selected due to the field of view of the camera (see Table 1). For closer distances the target body would be no longer completely within the field of view of the camera and no measurement can be produced by the sensor.

We can observe how the filter estimates improve during the approach. The second phase is executed with 1 cm/s guidance approach velocity. This is chosen because of safety considerations: the closer the distance, the smaller the relative velocity, to have sufficient reaction time in case of a malfunction.

Considering the eA-component of the attitude, see Fig. 22 we observe a bit more than 2 full rotations of the target around its spinning axis during the HiL test. Also

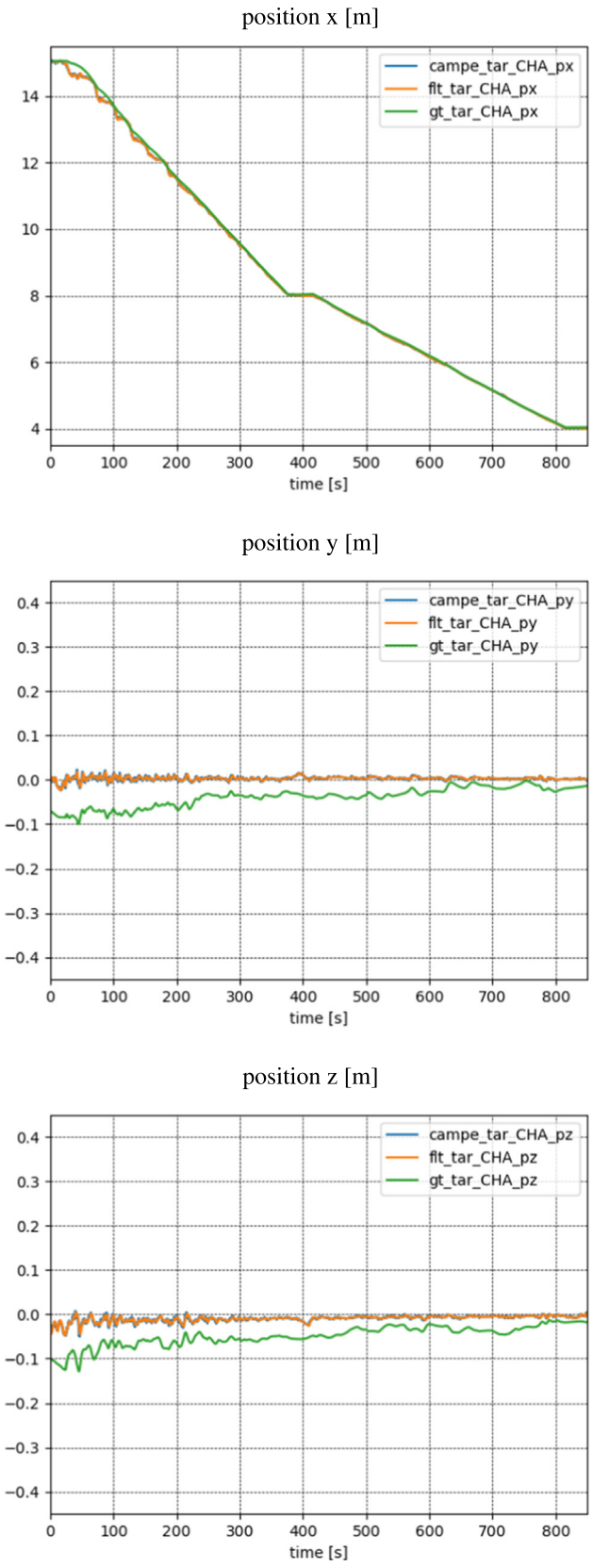
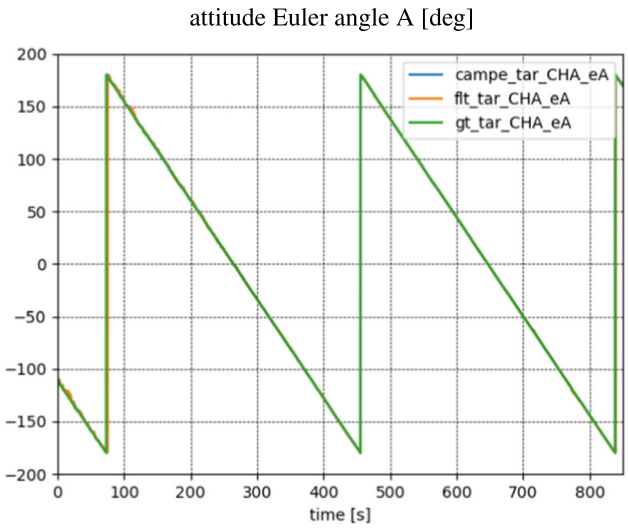
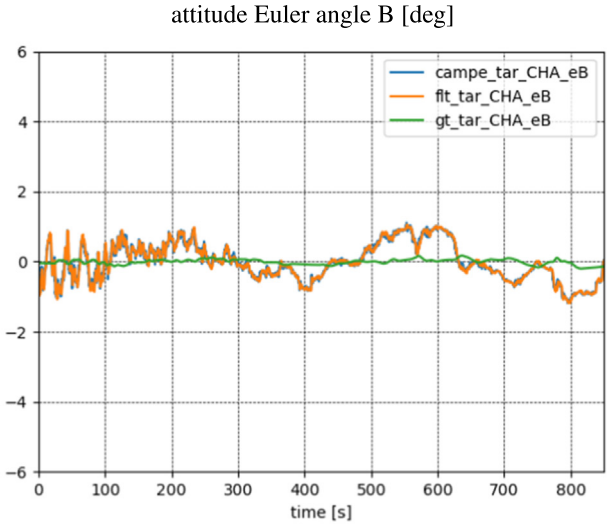


Fig. 21. Performance evaluation during an approach from 15 m to 4 m using a monocular camera as sensor - absolute position in chaser body frame (campe = camera pose estimation, flt = filter, gt = ground truth).





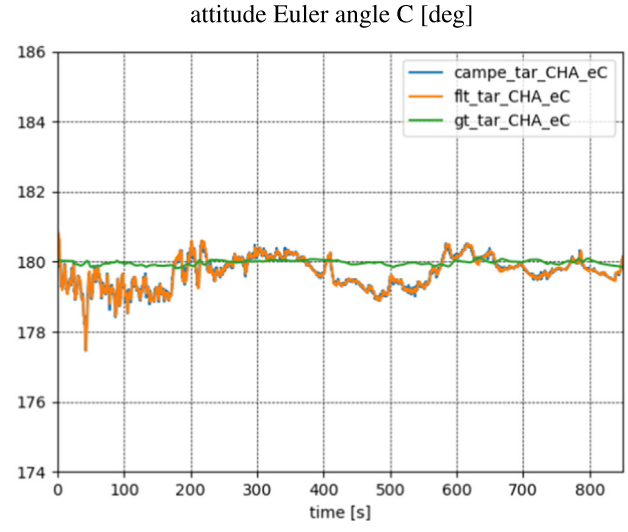


Fig. 22. Performance evaluation during an approach from 15 m to 4 m using a monocular camera as sensor - absolute attitude in chaser body frame (Euler angles in degrees, convention 1–2-3) (campe = camera pose estimation, flt = filter, gt = ground truth).

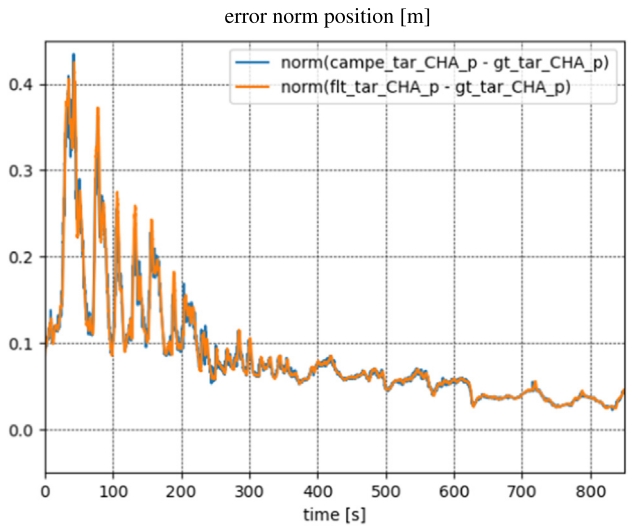


Fig. 23. Performance evaluation during an approach from 15 m to 4 m using a monocular camera as sensor - error norm of the position in chaser body frame (campe = camera pose estimation, flt = filter, gt = ground truth).

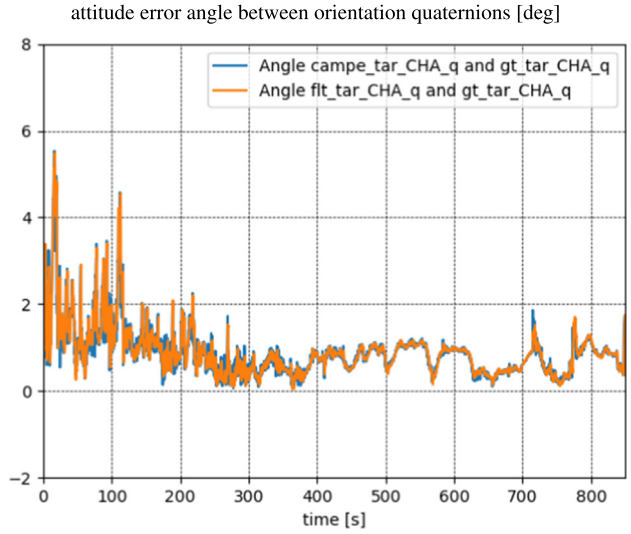


Fig. 24. Performance evaluation during an approach from 15 m to 4 m using a monocular camera as sensor - attitude error in chaser body frame (absolute angle between the compared orientations) (campe = camera pose estimation, flt = filter, gt = ground truth).

Table 5
Overview on test conditions for case 4 (approach with sensor fusion), coordinates of the target in chaser body frame (Euler angle convention 321).

	Initial	During test case
p_x [m]	15	15 to 8 with 2 cm/s, 8 to 4 with 1 cm/s
p_y [m]	0	approx. constant
p_z [m]	0	approx. constant
e_A [deg]	125	spinning 1 deg/s
e_B [deg]	0	approx. constant
e_C [deg]	180	approx. constant

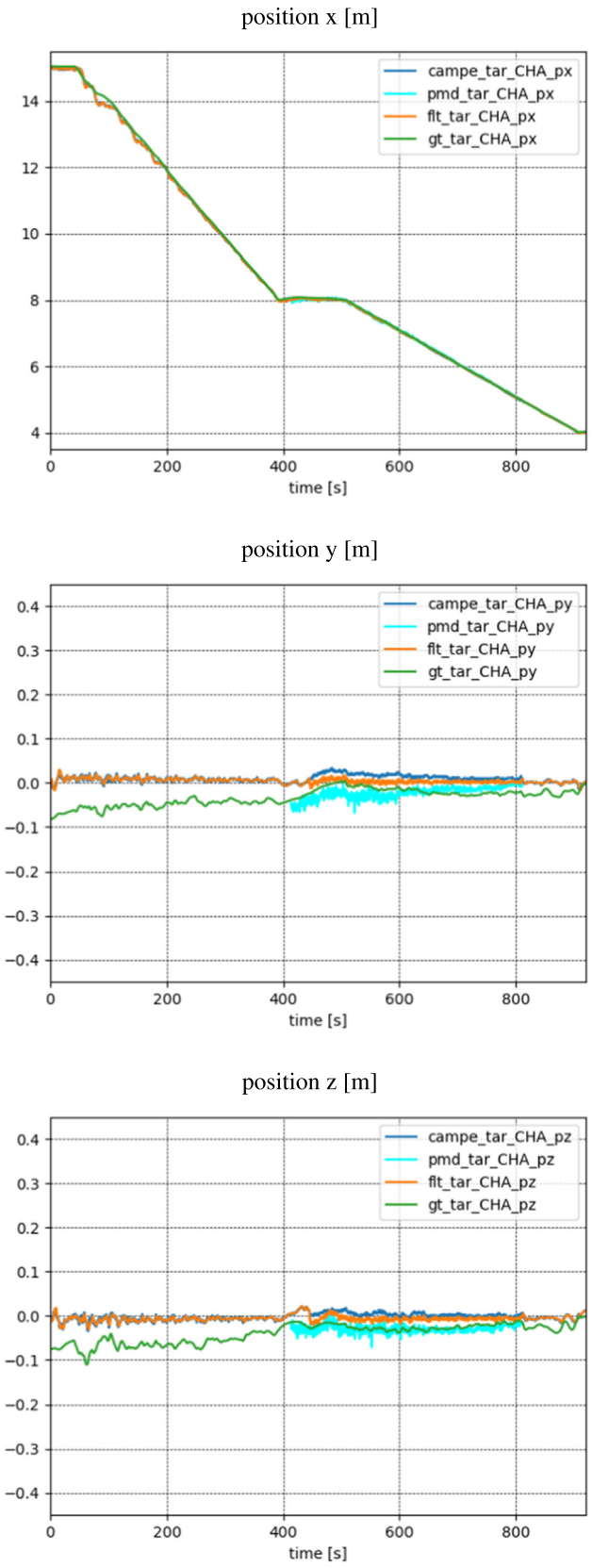


Fig. 25. Performance evaluation during an approach from 15 m to 4 m using sensor fusion - absolute position in chaser body frame (campe = camera pose estimation, pmdpe = PMD pose estimation, flt = filter, gt = ground truth).

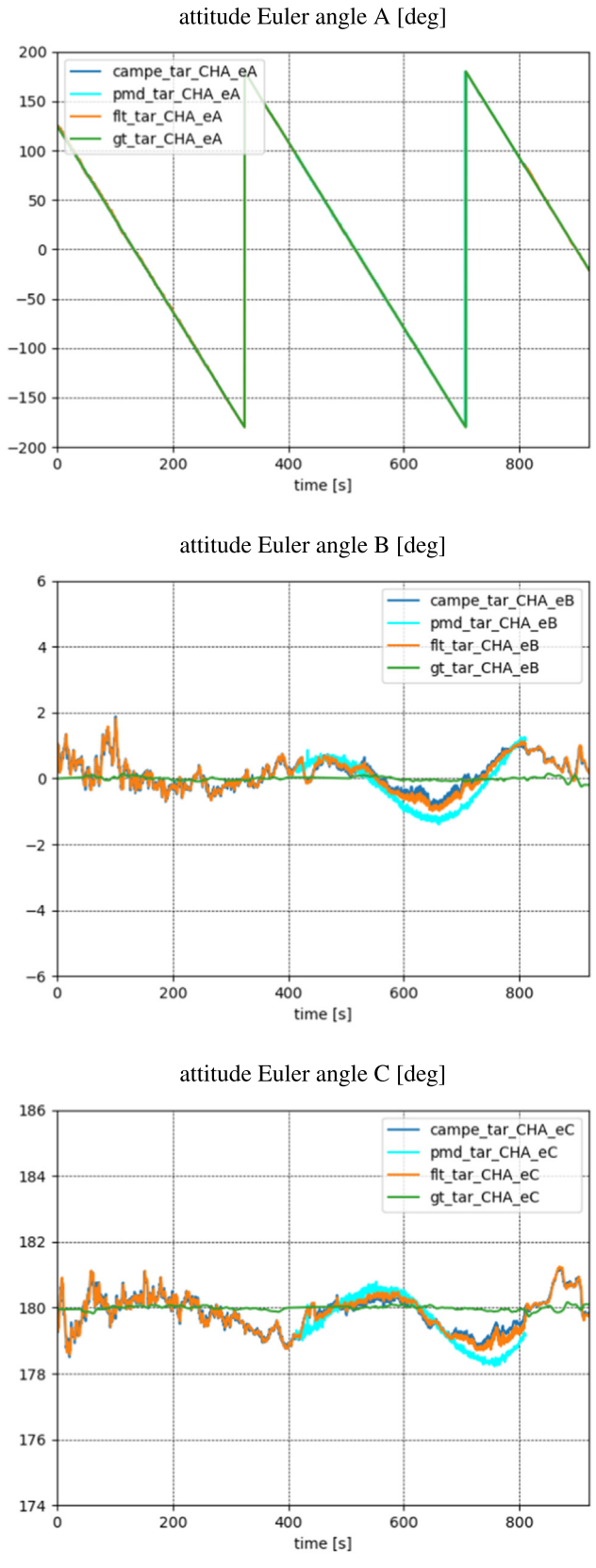


Fig. 26. Performance evaluation during an approach from 15 m to 4 m using sensor fusion - absolute attitude in chaser body frame (Euler angles in degrees, convention 1–2-3) (campe = camera pose estimation, pmdpe = PMD pose estimation, flt = filter, gt = ground truth).

the oscillations of the attitude errors decrease the closer we approach, see Fig. 24.

For both position and attitude, a decrease of the errors can be observed with decreasing distance to the target.

4.4. Case 4: approach with sensor fusion

As described in Section 2.2 the filter can perform sensor fusion in its filter update step. We therefore demonstrate an exemplary approach where two sensors are used: the monocular camera and the PMD camera, see Fig. 7.

Table 5 shows the initial and test conditions for this test case.

Fig. 25 and Fig. 26 show the absolute values of the position and attitude estimates, the measurements of monocular camera and PMD camera as well as the ground-truth of the EPOS robots for comparison. All values have been transformed to the chaser body system as common reference frame. The operational range of the PMD camera is from 8 m to 5 m, as explained in Section 3.2. Therefore only the phase between 8 m and 5 m can be performed with sensor fusion, the rest is done with the monocular camera only.

It can be well seen in Fig. 25 and Fig. 26 how the filter reacts when a second sensor is used and when sensor fusion is performed (see for example y- and z-coordinate of the position, or eB- and eC-component of the attitude): In the first phase (up to approximately 400 s of simulation time) the filter uses only camera measurements. As soon as a second measurement is available the filter estimate lies between the two sensor measurements. The estimates improves and the controller reacts. Therefore also the ground truth changes when a second measurement is used by the filter. The deviation of the ground truth from the guidance value (here both 0 for the y- and z-component) can be regarded as the control error.

Fig. 27 shows the error norm of the filter position estimate and the error norm of the sensor measurements from both sensors, monocular camera and PMD. Similarly, Fig. 28 shows the attitude error.

Again, we have a look at the two bitmasks *active sensors* and *used sensors*. The monocular camera is active for the entire approach, whereas the PMD camera is activated later at the intermediate hold point at 8 m. Fig. 29 visualizes the bitmask values. We choose the time span between 444 s and 448 s. At simulation time 445.4 s the PMD sensor was activated and first used by the filter at 445.5 s.

In Fig. 29 different combinations of new measurements can be observed.

- two new measurements are available (for example at time 445.5 s),
- no new measurement is available and the filter propagates only (at 445.6s),
- a new 2D camera measurement is available and no PMD measurement (at 445.9 s),

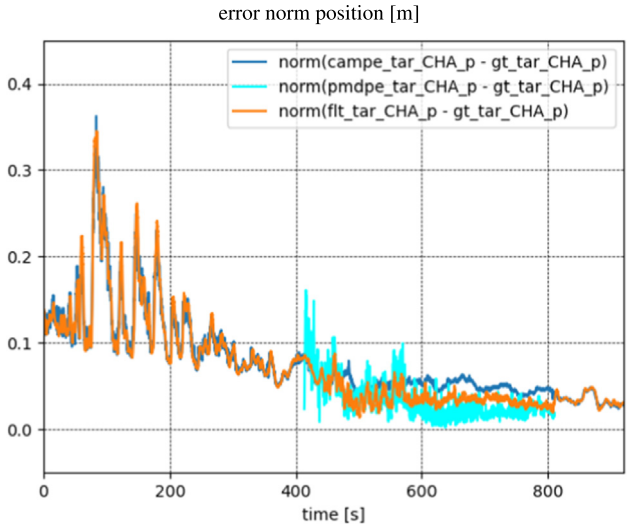


Fig. 27. Performance evaluation during an approach from 15 m to 4 m using sensor fusion - error norm of the position in chaser body frame (campe = camera pose estimation, pmdpe = PMD pose estimation, flt = filter, gt = ground truth).

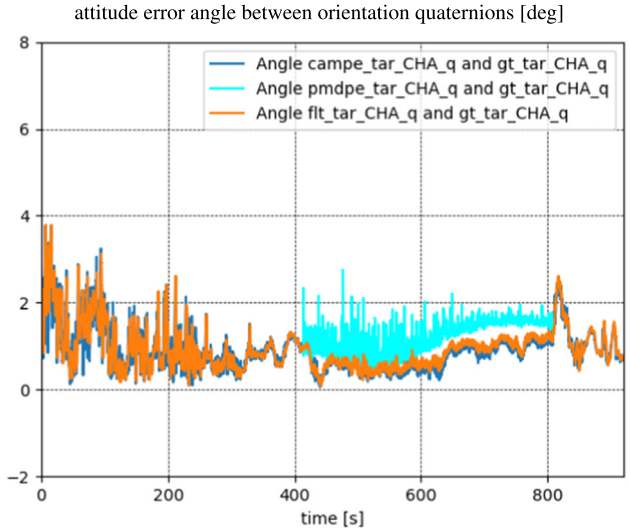


Fig. 28. Performance evaluation during an approach from 15 m to 4 m using sensor fusion - attitude error in chaser body frame (absolute angle between the compared orientations) (campe = camera pose estimation, pmdpe = PMD pose estimation, flt = filter, gt = ground truth).

- a new PMD measurement is available and no 2D camera measurement (at 446.0 s).

Further, the measurement frequencies do not have to be constant. This can be observed looking at the PMD measurements. Sometimes there is one filter execution with no new PMD measurement, sometimes there are two such filter executions.

Finally, we investigate the time delay of the sensor measurements visualized in Fig. 30 for the entire approach. The time synchronization and triggering of the pose estimation in the GNC system is beyond the scope of the paper, but

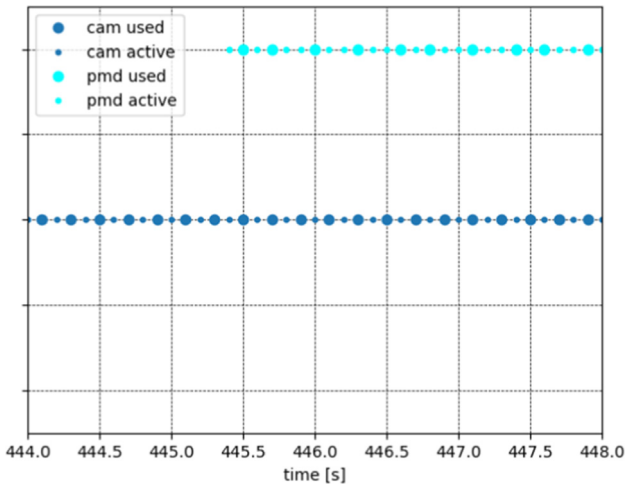


Fig. 29. Approach from 15 m to 4 m using sensor fusion - illustration of active and used sensors at simulation time [444, 448].

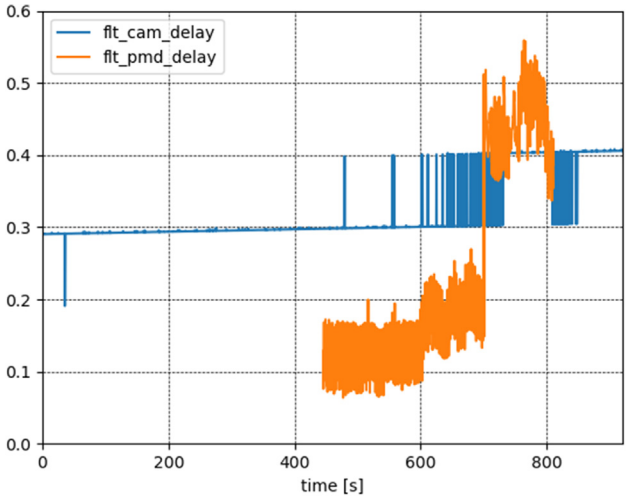


Fig. 30. Approach from 15 m to 4 m using sensor fusion - time delay [s] of the measurements.

we give some short explanations needed for analysis of Fig. 30. The camera pose estimation is triggered in fixed time steps. If no new camera frame can be received one time step is paused. This is why we see sometimes some alternation between 0.3 and 0.4 s of time delay. At the beginning the target's size in the camera image is smaller and the image processing is faster compared to later stages. When we are closer at the target, the computational time of the image processing increases. There is an intermediate phase where some images can be finished within 0.3 s, some other images need a bit longer, so one triggering misses the next frame. This is why some jumps in the camera delay can be observed. However alternating time delays do not lead to instability of the filter.

The PMD pose estimation is implemented in a different way. Here we see more noise. The processing time also increases the closer we approach to the target. We can observe one significant jump around 700 s from about 0.2 s to 0.4 s delay.

Table 6
Overview on test conditions for case 5 (approach with ScOSA OBC), coordinates of the target in chaser body frame (Euler angle convention 321).

	Initial	During test case
p_x [m]	15	15 to 8 with 2 cm/s, 8 to 4 with 1 cm/s
p_y [m]	0	approx. constant
p_z [m]	0	approx. constant
e_A [deg]	−20	spinning 1 deg/s
e_B [deg]	0	approx. constant
e_C [deg]	180	approx. constant

This example demonstrates that quite different types of delays can occur in practice. However the filter is able to generate stable pose estimates throughout the entire approach.

4.5. Case 5: single sensor during an approach with on-board computer hardware

In Section 3.4 the ScOSA ARM-based OBC has been presented which we use to test how the filter reacts on more realistic delays. The image processing routines needed for the camera measurements are computationally intensive. It is important to analyze the performance of the filter with the GNC system, including the filter and the image processing, executed on realistic on-board computer hardware.

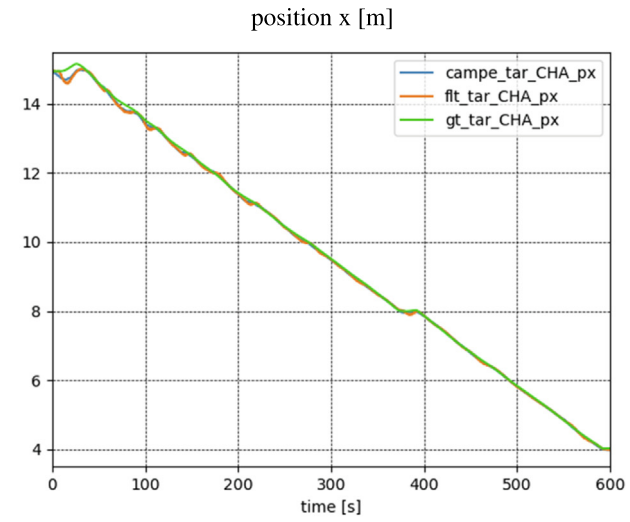
We therefore perform another approach from 15 m to 4 m with the GNC system executed on the ScOSA OBC.

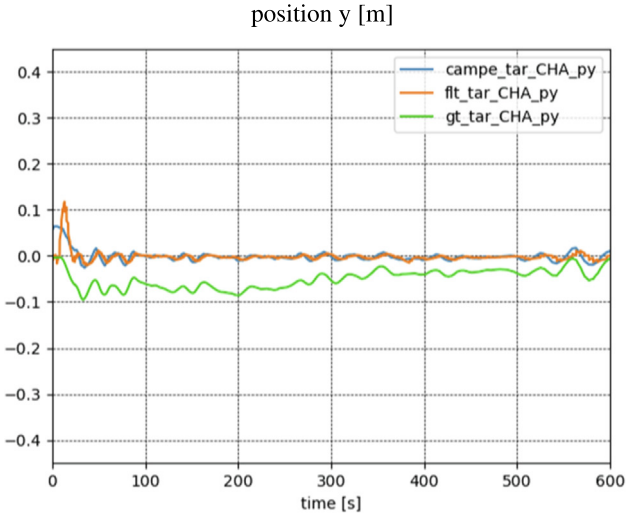
Table 6 shows the initial and test conditions for this test case.

Fig. 31 and Fig. 32 show the results of the position and attitude estimation. Fig. 33 and Fig. 34 present the position error and the attitude error. Using the ARM-based OBC the oscillations are higher compared to the previous test cases. However the error values decrease, the closer we approach the target (see for example Fig. 33).

The effect of the OBC can be well seen by Fig. 35 which illustrates the bitmasks *active sensors* and *used sensors*. Two different time spans are selected, at the beginning of the approach at time [60, 68] and at the end of the approach [540, 548]. Using a personal computer (x86 processor) the camera measurement delivers a new measurement approximately 5 times per second, compare Fig. 15. With the OBC we have only one measurement each 1.8 s at the beginning of the approach. As discussed several times in the paper, the image processing time increases during the approach to the target because of the increasing size of the target in the camera image. At the end of the approach we have only one measurement approximately every 5 s. This is a much lower measurement rate compared to the previous test cases. The use of the ScOSA OBC is a very challenging example and proves that the new filter method presented in this paper is able to cope with very low measurement frequencies.

The time delays using the ScOSA OBC (see Fig. 36) are much larger compared to the results presented in Fig. 16





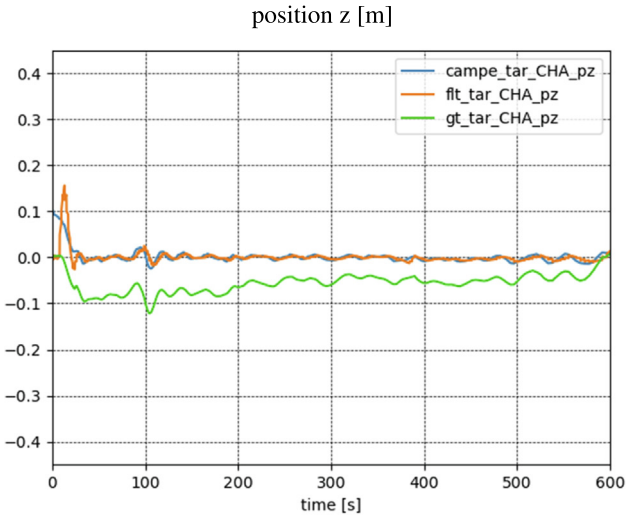
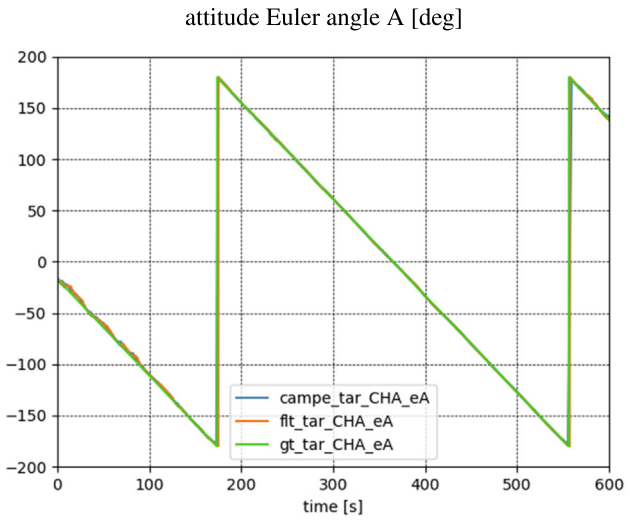
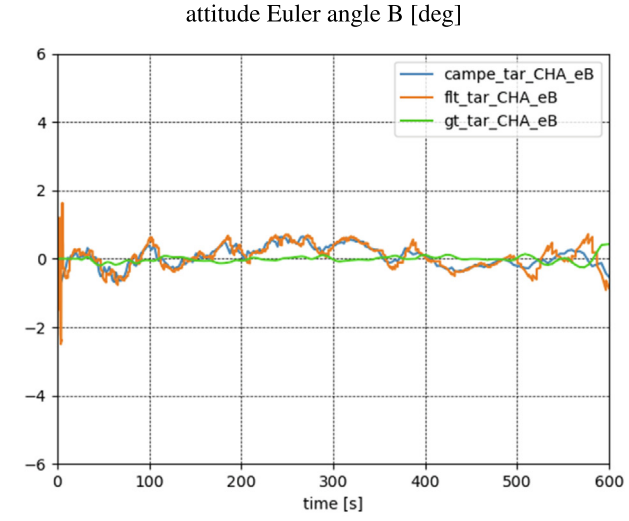


Fig. 31. Performance evaluation during an approach from 15 m to 4 m using the ScOSA OBC - absolute position in chaser body frame (campe = camera pose estimation, flt = filter, gt = ground truth).





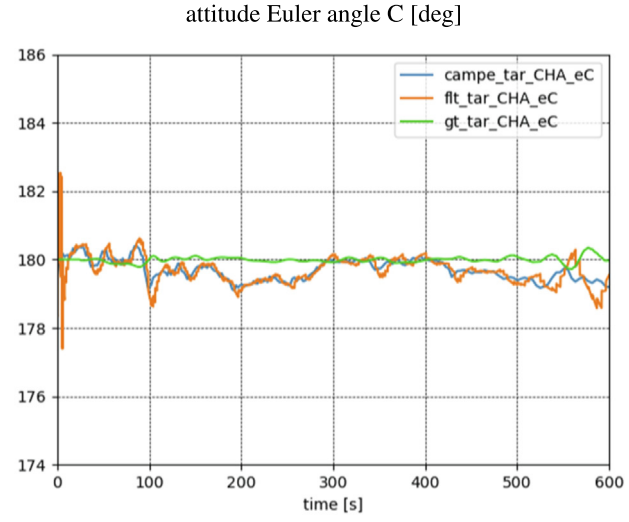


Fig. 32. Performance evaluation during an approach from 15 m to 4 m using the ScOSA OBC - absolute attitude in chaser body frame (Euler angles in degrees, convention 1–2-3) (campe = camera pose estimation, flt = filter, gt = ground truth).

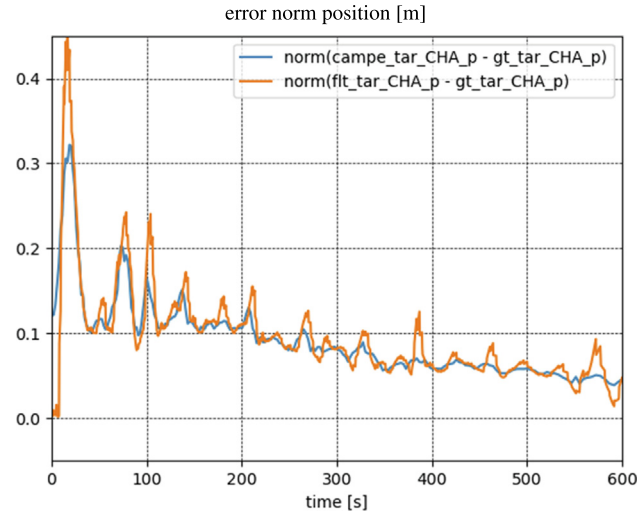


Fig. 33. Performance evaluation during an approach from 15 m to 4 m using the ScOSA OBC - error norm of the position in chaser body frame (campe = camera pose estimation, flt = filter, gt = ground truth).

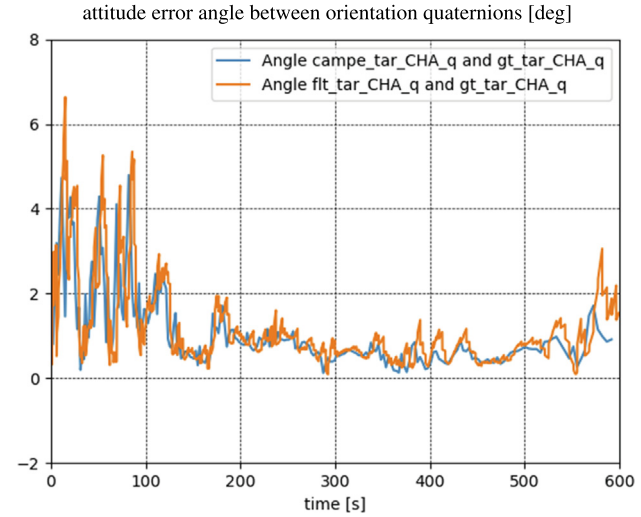


Fig. 34. Performance evaluation during an approach from 15 m to 4 m using the ScOSA OBC - attitude error in chaser body frame (absolute angle between the compared orientations) (campe = camera pose estimation, flt = filter, gt = ground truth).

and Fig. 30. The time delay at 15 m using the OBC, i.e. at the beginning of the approach, is around 2 s, which is already about 10 times larger than the time delay we observed in Fig. 16. The time delay continuously increases with increasing computational load for the image processing. At the end of the approach at 4 m distance the delay is even around 5 s. This observation matches with the frequency of the measurements that we observed in Fig. 35.

This demonstration shows that the filter is robust enough with respect to extreme measurement delays. In all test cases that we analyzed, it stays in a stable state, it does not diverge and provides a continuous state estimation during the entire rendezvous test.

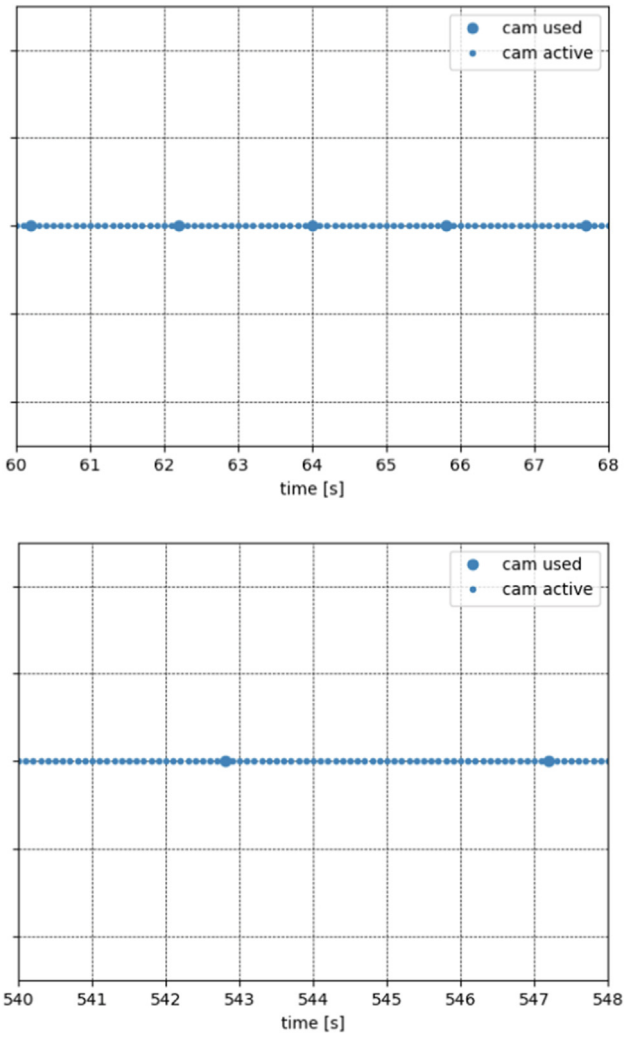


Fig. 35. Approach from 15 m to 4 m using the ScOSA OBC - - illustration of active and used sensors at simulation time [60, 68] (top) and [540, 548] (bottom).

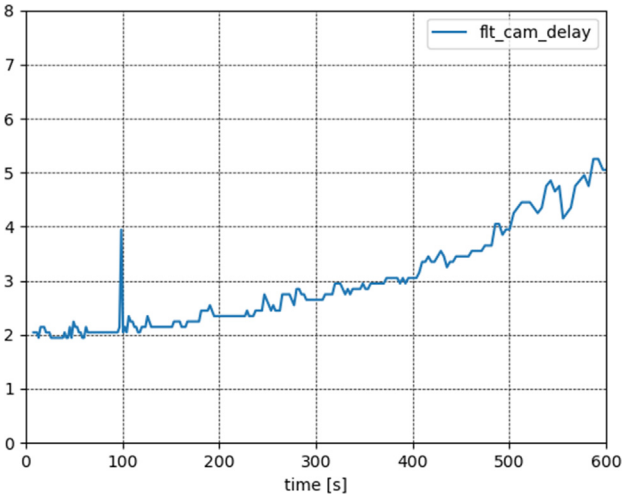


Fig. 36. Approach from 15 m to 4 m using the ScOSA OBC - time delay [s] of the camera measurements.

5. Discussion and conclusion

The results (see Section 4) demonstrate that the new navigation filter and thus the resulting control loop are stable for all presented test cases. As expected, the filter smoothes the sensor measurements and the performance improves the closer we approach.

The filter is able to perform sensor fusion. We tested the fusion with measurements from a 2D monocular camera and a 3D PMD camera. In principle, all kind of navigation sensors (see Section 1.1) can be integrated. In our test setup, the PMD sensor could only be used in the range between 5 m and 8 m distance to the target. In the future, we plan to replace the PMD camera by a LiDAR and want to analyze the performance of the filter in sensor fusion mode during the entire approach, not only during a subphase. Furthermore, if possible, we would like to test the filter on a real spaceflight mission or to compare the filter's performance with existing flight results of others.

In this paper, we showed that even for large measurement delays in the magnitude of 2-5 s, which occurred during the test with the ScOSA OBC, the filter provides stable state estimates. As expected the large delay led to higher oscillations and higher errors compared to the cases with small or moderate time delay. Some adaptations of the pose estimation could be done to reduce the delay. The current pose estimation routine is not yet optimized or parallelized.

This paper focuses on setting up a first version of a navigation filter which works well for a number of different test cases. Beyond the scope of the paper is filter tuning: For example, the performance of the filter is dependent on the system and measurement noise covariances Q_k and R_k (see Section 2.1.2) which can be tuned. Also the initial estimate $\overrightarrow{x}_0^{est}$ significantly influences how quick the filter converges at the beginning. In case of close range rendezvous, if information from a previous mid range rendezvous phase is known, this information should be used. If no information is available, the position and velocity of the chaser in ECI seemed to be a good guess for the position of the target according to tests done by us. The other components of the state, attitude and attitude rate, converged fast within a few seconds for arbitrary initial parameters like unit quaternion for the orientation or zero-3D vector for the rate. Additionally, also different sample times of the filter and different step sizes for the propagation should be tested and tuned. For all tuning parameters, one can perform for example Monte Carlo simulations to get some statistical results about the accuracy and stability of the filter. Based on that, the parameter setting can be tuned. The setup of such Monte Carlo simulations and the evaluation of the results was however not done in the scope of the paper, but is important for future research work.

Asynchronous measurements will become more and more important with more intelligent sensors and sensor processing methods involved (machine vision). For example, if certain low level pre-processing is integrated, the filter might receive asynchronous data of events. Furthermore, extensive real-time processing in connection with growing usage of multicore on-board computers require filter techniques that can cope with non-synchronized measurement output of the different cores. Apart of sensor fusion and delay handling, one major advance of the filter method is the handling of asynchronous measurements.

In summary, the new filter method seems to be a promising advancement compared to existing works (Larsen et al., 1998; DeKock et al., 2008; Zhang et al., 2014; Benninghoff et al., 2014). It could cope with a variety of test cases with different delays, distances and sensors noise characteristics.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Bar-Shalom, Y., 2002. Update with out-of-sequence measurements in tracking: exact solution. IEEE Trans. Aerosp. Electron. Syst. 38, 769–777. https://doi.org/10.1109/TAES.2002.1039398.

Benninghoff, H., Rems, F., Boge, T., 2014. Development and hardware-in-the-loop test of a guidance, navigation and control system for on-orbit servicing. Acta Astronaut. 102, 67–80. https://doi.org/10.1016/j.actaastro.2014.05.023.

Benninghoff, H., Rems, F., Risse, E.-A., et al., 2017. European Proximity Operations Simulator 2.0 (EPOS) - a robotic-based rendezvous and docking simulator. Journal of Large-Scale Research. Facilities 3, A107. https://doi.org/10.17815/jlsrf-3-155.

Benninghoff, H., Rems, F., Risse, E.-A., et al., 2018. End-to-end simulation and verification of GNC and robotic systems considering both space segment and ground segment. CEAS Space J. 10, 535–553. https://doi.org/10.1007/s12567-017-0192-2.

Burri, B., Frei, H., Rems, F. et al., 2021. Mars sample return - test campaign for near range image processing on European Proximity Operations Simulator. In: 11th International ESA Conference on Guidance, Navigation & Control Systems. Virtual.

DeKock, B.K., Betts, K.M., McDuffie, J.H. et al., 2008. Embedded relative navigation sensor fusion algorithms for autonomous rendezvous and docking missions. In: Proceedings of the 2008 National Technical Meeting of The Institute of Navigation, San Diego, California, U.S.A. pp. 193–203.

Fehse, W., 2003. Automated Rendezvous and Docking of Spacecraft. Cambridge Aerospace Series, Washington, D.C.

Grewal, M.S., Andrews, A.P., 2010. Applications of Kalman filtering in aerospace 1960 to the present [historical perspectives]. IEEE Control Syst. Mag. 30, 69–78. https://doi.org/10.1109/MCS.2010.936465.

Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. Trans. ASME, J. Basic Eng. 83, 35–45.

Kim, S.-G., Crassidis, J.L., Cheng, Y., et al., 2007. Kalman filtering for relative spacecraft attitude and position estimation. J. Guidance Control Dyn. 30, 133–143. https://doi.org/10.2514/1.G000204.

Klionovska, K., Burri, M., 2021. Hardware-in-the-loop simulations with umbra conditions for spacecraft rendezvous with PMD visual sensors. Sensors (Basel, Switzerland) 21, 1455. https://doi.org/10.3390/s21041455.

Klionovska, K., Venture, J., Benninghoff, H., et al., 2018. Close range tracking of an uncooperative target in a sequence of photonic mixer

device (PMD) images. Robotics 7, 5. https://doi.org/10.3390/robotics7010005.

Larsen, T.D., Andersen, N.A., Ravn, O. et al., 1998. Incorporation of time delayed measurements in a discrete-time Kalman filter. In: Proceedings of the 37th IEEE Conference on Decision and Control, Tampa, Florida, U.S.A. vol. 4. pp. 3972–3977.

Lund, A., Haj Hammadeh, Z.A., Kenny, P., et al., 2022. ScOSA system software: the reliable and scalable middleware for a heterogeneous and distributed on-board computer architecture. CEAS Space J. 14, 161–171. https://doi.org/10.1007/s12567-021-00371-7.

Marchand, E., Chaumette, F., Chabot, T. et al., 2019. RemoveDebris vision-based navigation preliminary results. In: IAC 2019-70th International Astronautical Congress, Washington D.C., U.S.A. pp. 1–10.

Mühlbauer, Q., Rank, P., Kaiser, C., 2013. On-ground verification of VIBANASS (vision based navigation sensor system): Capabilities and results. In: Proc. 12th Symposium on Advanced Space Technologies in Robotics and Automation. Noordwijk, The Netherlands.

Persson, S., Bodin, P., Gill, E. et al., 2006. PRISMA - an autonomous formation flying mission. In: Proc. Small Satellite Systems and Services - The 4S Symposium. Chia Laguna Sardinia, Italy.

Rems, F., Frei, H., Risse, E.-A., et al., 2021. 10-year anniversary of the European Proximity Operations Simulator 2.0 - looking back at test campaigns, rendezvous research and facility improvements. Aerospace 8, 235. https://doi.org/10.3390/aerospace8090235.

Simon, D., 2006. Optimal State Estimation - Kalman, H-infinity and Nonlinear Approaches. Wiley - Interscience, Hoboken, New Jersey.

Stoer, J., Bulirsch, R., 2002. Introduction to Numerical Analysis, vol. 3. Springer, New York, Berlin, Heidelberg.

Treudler, C.J., Benninghoff, H., Borchers, K. et al., 2018. ScOSA-scalable on-board computing for space avionics. In: Proceedings of the International Astronautical Congress, IAC. Bremen, Germany.

Tzschichholz, T., Boge, T., Schilling, K., 2015. Relative pose estimation of satellites using PMD-/CCD-sensor data fusion. Acta Astronaut. 109, 25–33. https://doi.org/10.1016/j.actaastro.2014.12.010.

Volpe, R., Circi, C., Sabatini, M., et al., 2022. GNC architecture for an optimal rendezvous to an uncooperative tumbling target using passive monocular camera. Acta Astronaut. 196, 380–393. https://doi.org/10.1016/j.actaastro.2020.10.038.

Volpe, R., Palmerini, G., Sabatini, M., 2017. Monocular and LIDAR based determination of shape, relative attitude and position of a non-cooperative unknown satellite. In: Proc. International Astronautical Congress (IAC). Adelaide, Australia.

Wertz, J.R., 2002. Attitude Determination and Control. Kluwer Academic Publishers, Dordrecht, Boston, London.

Zarchan, P., Musoff, H., 2000. Fundamentals of Kalman Filtering: A Practical Approach volume 190. Cambridge, Massachusetts: Progress in Astronautics and Aeronautics.

Zhang, K., Li, X.R., Zhu, Y., 2005a. Optimal update with out-of-sequence measurements. IEEE Trans. Signal Process. 53, 1992–2004. https://doi.org/10.1109/TSP.2005.847830.

Zhang, L., Yang, H., Zhang, S., et al., 2014. Kalman filtering for relative spacecraft attitude and position estimation: A revisit. J. Guidance Control Dyn. 37, 1706–1711. https://doi.org/10.2514/1.G000204.

Zhang, P., Gu, H., Milios, E.E., 2005b. Navigation with IMU/GPS/digital compass with unscented Kalman filter. In: IEEE International Conference Mechatronics and Automation 3, pp. 1497–1502. https://doi.org/10.1109/ICMA.2005.1626777.

Zhang, Q., 2000. Hybrid filtering for linear systems with non-Gaussian disturbances. IEEE Trans. Autom. Control 45, 50–61. https://doi.org/10.1109/9.827355.

Zhang, S., Bar-Shalom, Y., 2012. Optimal update with multiple out-of-sequence measurements with arbitrary arriving order. IEEE Trans. Aerosp. Electron. Syst. 48, 3116–3132. https://doi.org/10.1109/TAES.2012.6324681.

Zivan, Y., Choukroun, D., 2021. Dual quaternion Kalman filtering and observability analysis for satellite relative navigation with line-of-sight measurements. IEEE Trans. Aerosp. Electron. Syst. 58, 754–765. https://doi.org/10.1109/TAES.2021.3115584.