Vision-based Gyroscope Fault Detection for UAVs

Benedict Simlinger, Guillaume Ducard

This full text paper was peer-reviewed at the direction of IEEE Instrumentation and Measurement Society prior to the acceptance and publication.

This is the accepted version of : B. Simlinger, G. Ducard, "Vision-based Gyroscope Fault Detection for UAVs," in Proceedings of the 2019 IEEE Symposium on Sensor Applications, Sophia Antipolis, France, March 11 – 13, 2019. https://doi.org/10.1109/SAS.2019.8705965

Vision-based Gyroscope Fault Detection for UAVs

Benedict Simlinger, Guillaume Ducard

***Abstract*—This paper presents a vision-based fault detection and isolation architecture for unmanned aerial vehicles. The vehicle's attitude is computed from visual input through a horizon tracking algorithm, independently of any other sensor. In a second stage, two Kalman filters are used for fault detection and identification in two gyroscopes. The loosely coupled architecture is suitable for real-time application. The algorithm was implemented with the ROS framework and the system's performance is evaluated in a real-time application scenario with artificially introduced sensor faults.**

***Index Terms*— Vision, gyroscope fault detection and isolation, unmanned aerial vehicles (UAVs), horizon tracking**

I. Introduction

SENSOR fault detection, isolation and mitigation is essential for the safe operation of Unmanned Aerial Vehicles (UAVs). Using vision for fault detection can be a favourable choice if the UAV already carries vision components for other purposes such as filming or inspection. With the camera already mounted on the UAV, vision-based fault detection and isolation (FDI) will induce little additional cost on the UAV in terms of weight or complexity, while introducing additional, independent sensory information to the FDI system in place.

This paper identifies three obstacles to be overcome for the successful application of vision-based FDI in the following order:

1) extract meaningful information from the vision input,
2) harmonize data representation and filter signals for conflicting sensory input,
3) apply a fault detection and identification algorithm.

Therefore, the proposed system combines aspects of vision processing and FDI, as presented in the following sections. A literature review follows in the next two paragraphs.

A. Attitude from Vision

There is an abundant body of literature on vision-based attitude estimation [1]. The majority of recent research in the area of attitude estimation is dedicated to multi-rotor helicopter drones. Vision-based attitude estimation approaches commonly depend on the input from additional sensors, such as gyroscopes, which forbids vision-based FDI for gyroscopes. Machine learning approaches have also emerged recently, trying to solve the problem of visual attitude estimation [2], but vision-based FDI is still an open problem.

B. Simlinger. E-mail: b.simlinger@gmx.at
G. Ducard is with the I3S CNRS/University of Nice Sophia Antipolis, France. E-mail: ducard@i3s.unice.fr

B. Kalman Filter based Fault Detection

Single Kalman filters can be used for fault detection such as presented in [3]. A wide variety of Kalman filter variants is available and the optimal choice of the Kalman filter variant depends on the application at hand [4] [5]. An other popular approach to FDI is the use of Multiple-Model Adaptive Estimation (MMAE), which is based on banks of Kalman filters [6].

Other approaches to FDI include statistical analysis of signals such as presented in [7]. More complex, high level sensor fault-tolerant architectures can reconstruct the data of deficient sensors as shown in [8] or use redundant sensor information in combination with a majority voting algorithms to obtain more reliable attitude information [9]. Some approaches provide fail-safe or explicitly robust sensor fusion algorithms, where redundant sensors are used as drop-in replacement in case of sensor failure [4]. The work in [10] uses three redundant gyroscopes for attitude estimation. This allows to identify a faulty sensor through a majority vote-like mechanism based on the Mahalanobis distance, but requires additional preprocessing of the signals.

To the best of the author's knowledge, there exist little to no research on vision-based FDI for gyroscopes [11] [12].

The contribution of this paper is to demonstrate the use of an independent vision-based attitude estimate for the purpose of FDI in gyroscopes. The attitude estimates are derived from an on-board camera which is pre-installed on a UAV for mission specific tasks. An efficient computational procedure is derived to segment images for faster horizon tracking. The FDI algorithm is based on Kalman filters. This approach circumvents the need to install an additional third gyroscope on the UAV and to implement a dedicated majority voting system such as presented in [10]. This architecture for FDI of roll and pitch gyroscopes can prevent fatal crashes of the vehicle during autonomous flight.

II. Fault Detection and Isolation Architecture

The architecture suggested in this paper detects and isolates sensor faults in gyroscopes in real-time with the help of visual queues obtained from an on-board camera. The underlying idea can be likened to a human pilot visually validating sensory measurements by observing the outside world.

This algorithm is limited to detecting and isolating sensory faults only. It is not tasked to mitigate the fault or to make high-level decision based on the fault diagnosis.

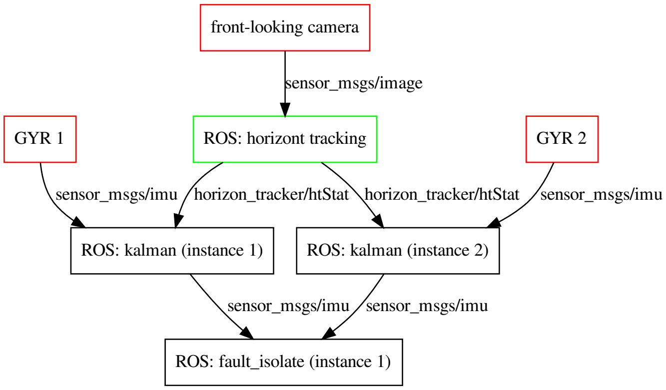

Fig. 1. Schematic of the vision-based fault detection algorithm. The system is provided with data from a camera and two gyroscopes (GYR1, GYR2). The ROS message formats are notated at the edges.

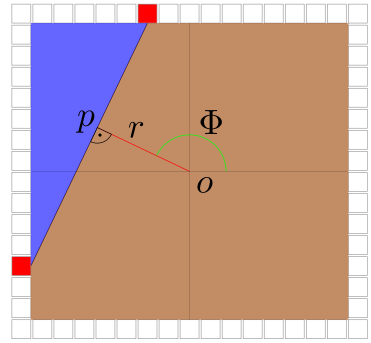

Fig. 2. The Hough Transformation maps a (horizon) line to a representation that is given as a point coordinate $p = (r, \Phi)$. The point p is the foot of the line to the origin o at a distance r (red). The angle Φ is between the camera horizontal line and the line segment $[op]$.

A. Structure of Architecture

The FDI architecture is made from several self-contained modules. As shown in Figure 1, the FDI system features a horizon tracker node, two Kalman filters for detecting faults and a fault isolation node for isolating faults in the UAV's gyroscopes.

The horizon tracker node estimates the roll and pitch angle of the UAV. It can not extract information about the yaw angle of the UAV, however this is sufficient for meaningful fault detection. Compared to a fault in yaw-rate measurements, faults in roll and/or pitch-rate measurements are more likely to lead to a fatal crash during autonomous flight since they influence the aircraft's orientation control drastically.

Kalman filters, as presented in subsection II-C, are used to fuse the information from the horizon tracker with the gyroscope measurement. Since the horizon tracker measures the attitude and the gyroscopes measure angular velocities, or rate of change of attitude, a Kalman filter is the ideal mechanism to fuse those two types of signal. For each gyroscope, there is one Kalman filter.

As shown in Figure 1 the output of each Kalman filter is processed by a fault isolation node, which is presented in subsection II-D. The output of the fault isolation node indicates whether there is a fault in the horizon tracker or if one of the two gyroscopes is faulty.

B. Horizon Tracking

To detect faults in gyroscopes, the airplane's attitude must be obtained from an independent source. In this work, attitude information is extracted from a video stream by observing the horizon line visible in the image frames without relying on additional sensory information such as inertial measurement unit (IMU) measurements. The horizon tracker algorithm presented in this work does not apply to images from omnidirectional cameras.

The frame rate of the video stream determines the update frequency of the Kalman filters. In this paper, a video frame rate of 10 Hz was used. If the frame rate it too low, the vision measurements do not keep up with the pace of the gyroscope measurements and the FDI system will interpret the discrepancy as sensor fault. The camera should feature a big field of view (FOV) and should be mounted in a forward-looking configuration. The horizon tracker computes current roll and pitch angles of the airplane's attitude relative to the world frame.

Note that the work in [2] is an approach that performs sky segmentation based on machine learning. It would be suitable as drop-in replacement for the sky-segmentation method used in this work, but the effort required to reproduce the results and the hardware requirements for running the segmentation algorithm are beyond the scope of this study.

1) Algorithm: In this work the algorithm used for the horizon tracker is based on previous research [13] and was adapted and improved upon. Most notably, the algorithm presented here deviates from [13] in the following points:

- The algorithm of this paper uses gray-scale input instead of a three channel RGB color image. As a consequence, the precautions made against illconditioned images in [13] become the default mode of operation in this paper.
- The paper [13] covers the complete search space for every frame, without taking into account that consecutive frames will have closely co-located horizon lines. Paper [13] is aware of that fact but decided against using this property because their video stream may be subject to transmission faults before it is processed on a ground station computer. This paper processes the video stream on board of the UAV.

The horizon line is represented by its *Hough Transform* and is specified by

$$p = (r, \Phi)$$

which denotes the distance r of the point p to the origin o and Φ as the angle of the vector $v = p - o$. The point p is the point on the horizon line that is closest to the origin o. The origin is located at the focus of expansion, which is described in section II-B3. The horizon line segments an image into two areas, the sky and the ground area.

The horizon tracker works on a frame-by-frame basis. Searching for the horizon line in a video image frame is limited to the vicinity of the previous horizon line under the assumption that the vehicle's angular velocities stay within

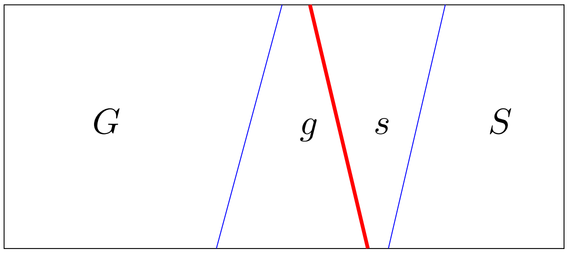

Fig. 3. Instead of segmenting the image (represented by the frame) into two areas representing ground and sky, the image is segmented into four areas. The blue lines represent the limits of search space for the horizon line of this frame. The red line within the search space limits is the currently evaluated horizon line. Area G will be evaluated as ground for all iterations of the search in a given frame. Area g changes during search. Areas S and s behave likewise. Avoiding re-evaluation of areas G and S reduces computational costs significantly, see section III-B.

predetermined limits. This assumption allows to reduce the required computations considerably.

All horizon lines in the search space are evaluated. The best line, according to the evaluation metric in Equation 1, is selected as the artificial horizon. In order to evaluate the fitness f of an image segmentation,

$$f = \frac{1}{var_s + var_g} \tag{1}$$

is used, where var_s denotes the variance of the gray-scale pixels' value that are located in the sky-area of the image and var_g denotes the variance of gray-scale pixels' value in the ground area. The horizon line with the maximum evaluation value f represents the most likely image segmentations that coincides with the real world horizon line.

In addition to limiting the search space for the horizon line, this work adds an other improvement. The pixels outside the horizon line search space will be considered as ground pixels or sky pixels for every iteration of the horizon search acting upon the same video frame. Only pixels inside the search space (refer to Figure 3) may be categorized as sky or ground depending on the currently evaluated horizon line. Splitting the image into four segments

- area outside the search space considered as ground: G
- area inside the search space considered as ground: g
- area inside the search space considered as sky: s
- area outside the search space considered as sky: S

helps to reduce calculations. Details on this approach are presented the following section.

2) Accelerating the Horizon Line Evaluation: The horizon tracker evaluates horizon lines according to Equation 1. Increasing the speed at which the variance terms can be calculated will increase the speed of the overall horizon search algorithm. Equation 2 below

$$var(x) = \frac{1}{n}\sum_{i=0}^{n} x_i^2 - \left(\frac{1}{n}\sum_{i=0}^{n} x_i\right)^2 = \overline{x^2} - \overline{x}^2 \tag{2}$$

presents the formulation of the sample covariance. The sample vector x of all frame pixels can now be split at index h into two vectors such that

$$x = [x_{i\le h}, x_{h<i}]. \tag{3}$$

In this work, such a split is used to divide the ground area of a frame into areas G and g, and the sky area into areas s and S, respectively. This separation enables the reuse of the calculations for G and S for every iteration step of the search algorithm, because G and S are outside the search space of the horizon line. In comparison, g and s need to be re-evaluated at every iteration step. The following equations are indexed for a one-dimensional array but are valid for two dimensional arrays, such as images, too:

$$\begin{aligned}
var(x) =& \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)^2 && (4)\\
=& \frac{1}{n}\sum_{i=1}^{h} x_i^2 + \frac{1}{n}\sum_{i=h+1}^{n} x_i^2 \\
& - \left(\frac{1}{n}\sum_{i=1}^{h} x_i + \frac{1}{n}\sum_{i=h+1}^{n} x_i\right)^2 && (5)\\
=& \frac{h}{n}\underbrace{\overline{x^2}_{1\le i\le h}}_{A} + \frac{n-h}{n}\underbrace{\overline{x^2}_{h+1\le i\le n}}_{B} \\
& - \left(\frac{h}{n}\cdot\underbrace{\overline{x}_{1\le i\le h}}_{C} + \frac{n-h}{n}\cdot\underbrace{\overline{x}_{h+1\le i\le n}}_{D}\right)^2 && (6)
\end{aligned}$$

We recognize that terms A and C correspond to evaluating area G or S. They will remain unchanged for every horizon search step in a frame. Reusing the calculated variance terms of G and S saves valuable CPU time. A quantification of the performance gain is given in subsection III-B.

Finally we need to find n and h in order to know all terms of Equation 6. Since n and h represent the number of pixels we can approximate those values by calculating the area of the image segments.

3) Attitude from Horizon Line Estimates: Knowing the horizon line alone does not give any direct information about the attitude of the UAV. The position of the horizon line in the image frames must be related to a body-fixed reference frame.

The focus of expansion is located at the origin of all optical flow during a straight-forward movement of the UAV with a forward-looking camera. The focus of expansion is visualized in Figure 4.

The distance r of the focus of expansion to the horizon line, shown in Figure 2 indicates the aircraft's pitch angle θ in world reference frame. Given an undistorted image, the conversion factor for converting r to θ can be deducted from the camera's angle of view α and its resolution ρ (in pixels) by $r/\theta = \rho/\alpha$. The angle between the horizon line and the wing plane of the UAV allows to intuitively deduce the roll angle ϕ of the UAV.

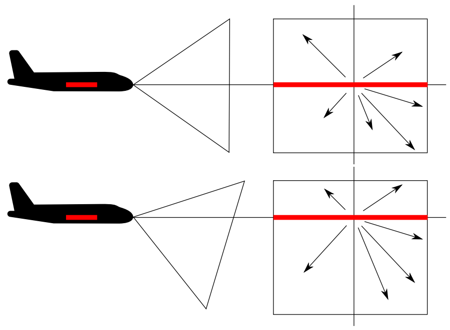

Fig. 4. Changing the angle of the camera mount influences the field of view and focus of expansion accordingly. The wing plane is marked red. The black arrows indicate the direction of the optical flow.

C. Kalman Filter based Fault Detection

In order to detect faults in the gyroscopes two Kalman filters are employed as shown in Figure 1. In this paper, angular velocity measurements from the gyroscopes are fused with attitude estimates from the horizon tracking module in a direct Kalman filter, driving the high-frequency prediction and low-frequency update steps of the Kalman filters, respectively. The state vector of the Kalman filter is made of the airplane's attitude Φ_{IB} (as quaternion) and the gyroscope's bias estimates b_ω.

$$x = (\Phi_{IB}, b_\omega), \in \mathbb{R}^7 \tag{7}$$

Angular velocity measurements (in body frame coordinates) from the gyroscopes are represented by the vector

$$\omega \in \mathbb{R}^3, \tag{8}$$

while the measurements from the horizon tracker are in the form of quaternions denoted

$$\Omega \in \mathbb{R}^4 \tag{9}$$

The following matrices are used for the implementation of the Kalman filter

- $F \in \mathbb{R}^{6\times6}$: State transition matrix, see Equation 22
- $Q \in \mathbb{R}^{6\times6}$: Process noise covariance matrix, derived from IMU datasheet
- $G \in \mathbb{R}^{6\times6}$: process noise transition matrix, see Equation 23
- $H \in \mathbb{R}^{3\times6}$: Observation model matrix
- $R \in \mathbb{R}^{3\times3}$: Observation noise covariance matrix
- $P \in \mathbb{R}^{6\times6}$: State estimate error covariance matrix
- $K \in \mathbb{R}^{6\times3}$: Kalman gain matrix

The dimension of the state vector seems inconsistent with the dimensions of the above matrices and Equation 10. These inconsistencies will be resolved through the operators $\boxplus$ and $\boxminus$ later on. For more details on those operators refer to [14].

The discrete formulation of the Kalman filter prediction step is:

$$x = x + \omega \Delta t \tag{10}$$
$$P = FPF^T + GQG^T \tag{11}$$

Update steps are performed according to:

$$K = PH^T(HPH + R)^{-1} \tag{12}$$
$$x = x + K(\Omega - Hx) \tag{13}$$
$$P = (I - KH)P(I - KH)^T + KRK^T \tag{14}$$

Note, that the state estimate update in Equation 10 is not based on the state transition matrix F, but on gyroscope measurements ω instead.

Instead of a fused state estimate, the Kalman filters publish their respective innovation terms $(\Omega - Hx)$ from Equation 13. An increased innovation term indicates a sensor fault in either gyroscope or horizon tracker. But it cannot determine which sensor, IMU or horizon tracker, is faulty. This task will be solved by the algorithm presented in subsection II-D.

A unit quaternion $\Phi_{BA} \in SO(3)$ represents a relative orientation of a coordinate system B w.r.t. to an other coordinate system A. An orientation can be modified by applying a rotation $\varphi \in \mathbb{R}^3$. Since the orientation is represented by a four-dimensional quaternion and the rotation to be applied is represented by a three-dimensional vector, *boxplus* $\boxplus$: $SO(3) \times \mathbb{R}^3 \rightarrow SO(3)$ is defined as follows [14] :

$$\boxplus : \Phi, \varphi \rightarrow exp(\varphi) \circ \Phi \tag{15}$$

The *boxminus* operator, $\boxminus : SO(3) \times SO(3) \rightarrow \mathbb{R}^3$, needs to be introduced for calculating the difference between two quaternions, which is a three-dimensional rotation:

$$\boxminus : \Phi_1, \Phi_2 \rightarrow log(\Phi_1 \circ \Phi_2^{-1}) \tag{16}$$

Both operators represent the application of functions that transform quaternions to rotations or vice versa and a concatenation function.

The concatenation $\circ : SO(3) \times SO(3) \rightarrow SO(3)$ of two quaternions yields an other quaternion and is evaluated as follows [14]:

$$\Phi_1 \circ \Phi_2 = (q_0 p_0 - \vec{q}^T \vec{p}, q_0 \vec{p} + p_0 \vec{q} + \vec{q} \times \vec{p}) \tag{17}$$

The exponential map $exp : \mathbb{R}^3 \rightarrow SO(3)$ represents the transformation of a rotation into an orientation.

$$exp(\varphi) = \quad (q_0, \vec{q}) = \left(cos(||\varphi||/2), sin(||\varphi||/2)\frac{\varphi}{||\varphi||}\right) \tag{18}$$
$$exp(\varphi) \approx \quad (1, \varphi/2), (||\varphi|| \approx 0) \tag{19}$$

The logarithm is the inverse function of the exponential map and transforms a relative orientation into its respective rotation vector:

$$log(\Phi) \quad = \quad 2atan2(||\vec{q}||, q_0)\frac{\vec{q}}{||\vec{q}||} \tag{20}$$
$$log(\Phi) \quad \approx \quad \vec{q}, (||\vec{q}|| \approx 0) \tag{21}$$

The Kalman filter matrices for the horizon tracking subsystem are calculated as:

$$F = \begin{bmatrix} I & -\Delta C(\Phi_{IB})\Gamma(\Delta t\omega) \\ 0 & I \end{bmatrix} \tag{22}$$

$$G = \begin{bmatrix} -\Delta t C(\Phi_{IB})\Gamma(\Delta t\omega) & 0 \\ 0 & \Delta t I \end{bmatrix} \tag{23}$$

TABLE I
VARIOUS SCENARIOS OF FAULTS.

case	vision	imu1	imu2	fault isolation
1	OK	OK	OK	
2	OK	OK	fault	yes
3	OK	fault	OK	yes
4	OK	fault	fault	no
5	fault	OK	OK	yes
6	fault	OK	fault	no
7	fault	fault	OK	no
8	fault	fault	fault	no

where Δt is the time step, $\Gamma(\varphi) \in \mathbb{R}^{3\times3}$ is the derivative of the exponential map

$$\Gamma(\varphi) \begin{cases} = & I + \frac{(1-cos(||\varphi||))\varphi^{\times}}{||\varphi||^2} + \frac{(||\varphi||-sin(||\varphi||))\varphi^{\times 2}}{||\varphi||^3} \\ \approx & I + 0.5\varphi^{\times}, ||\varphi|| \approx 0 \end{cases} \tag{24}$$

and $C : \mathbb{R}^3 \rightarrow \mathbb{R}^{3\times3}$ is a rotation matrix

$$C(\varphi) \begin{cases} = & I + \frac{(1-cos(||\varphi||))\varphi^{\times 2}}{||\varphi||^2} + \frac{sin(||\varphi||)\varphi^{\times}}{||\varphi||} \\ \approx & I + \varphi^{\times}, ||\varphi|| \approx 0 \end{cases} \tag{25}$$

Equation 24 and Equation 25 use the skew symmetric matrix of a vector $v \in \mathbb{R}^3$ denoted as $v^{\times}$. With above definitions, the mismatch of dimension is resolved, because $\boxplus$ allows to add three-dimensional gyroscope measurement to four-dimensional quaternions and $\boxminus$ is its inverse operator.

D. Fault Isolation

In this paper two gyroscopes and a single forward-looking camera provide the necessary inputs for fault isolation.

A fault is detected if the innovation term of a Kalman filter exceeds a empirically determined threshold. Since both Kalman filters share the same input from the vision system but fuse it with measurements from two distinct gyroscopes, the faulty sensor may be inferred. Possible fault scenarios are presented in Table I.

case 1:
: Cues from the vision system are in line with both gyroscopes. All sensor measurements are valid.

case 2 and 3:
: Exactly one Kalman filter exhibits innovation terms above a predefined fault indication threshold. The gyroscope connected to this Kalman filter is faulty.

case 4:
: The system detects a fault in both Kalman filter. If both gyroscopes exhibit the same fault, e.g. sensor measurements stuck to zero, the faulty gyroscopes will overrule the correct measurements from the vision system. This scenario can not be differentiated from case 5.

case 5:
: The system detects faults in both Kalman filters. If the measurements of the gyroscopes are similar, the fault can be isolated and the vision input will be recognized as faulty.

case 6, 7 and 8:
: The architecture will detect faults but can not isolate the fault(s). In all those cases, the majority of sensors is faulty.

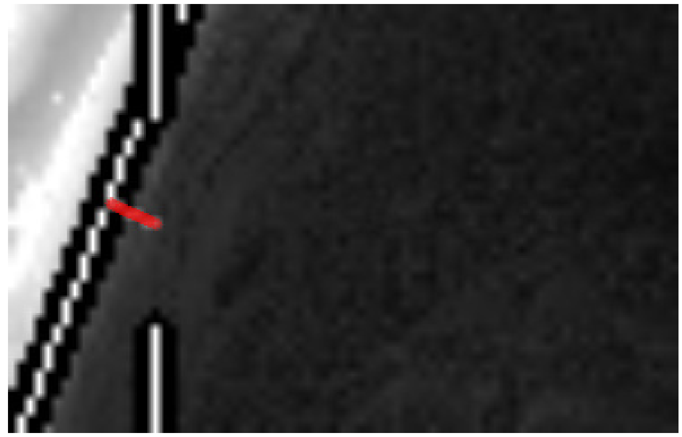

Fig. 5. Visual output of the horizon tracker. The interrupted vertical line represents the wing plane of the UAV. The continuous line visualizes the horizon estimate. The thin, red line indicates the distance of the focus of expansion to the horizon line.

III. RESULTS

A. Hardware Configuration

The work was tested on data recorded by the fixed-wing team of Autonomous Systems Lab from ETH Zürich [1]. In this section a recording system featuring one camera and 2 IMUs was used. The data was recorded on 25.7.2014 in Switzerland. The camera provided an image stream at 10 images per second. IMU measurements were provided by two different gyroscopes at 2.8 kHz and 49.5 Hz respectively.

Sensor faults did not occur during the experiments but were introduced artificially to the recorded experimental data, which was processed in real-time on an Intel Atom processor.

B. Attitude from Vision

The horizon tracker works under various day light conditions. Figure 5 shows an example frame of the video stream with the horizon line and the wing plane as overlay. However, it fails in cloudy conditions as shown in Figure 7.

The performance improvement presented in subsubsection II-B2 lower the processing time for one frame by a factor of

$$\frac{\frac{1}{h}(G+S)+g+s}{G+S+g+s} \tag{26}$$

where h denotes the video frame frequency, and G, S, g and s denote the areas according to Figure 3.

C. Fault Detection

Figure 8 displays the innovation terms of both Kalman filters. If the values surpass a predefined threshold value, the system identifies the faulty gyroscope. In this experiment, the sensor faults were introduced into the recorded gyroscope measurements.

[1] http://www.asl.ethz.ch/

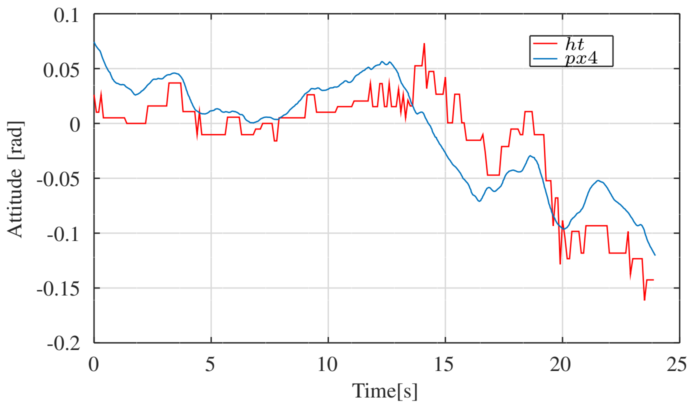

Fig. 6. Pitch angle estimated by horizon tracker (red) and PX4 flight controller (blue) for reference.

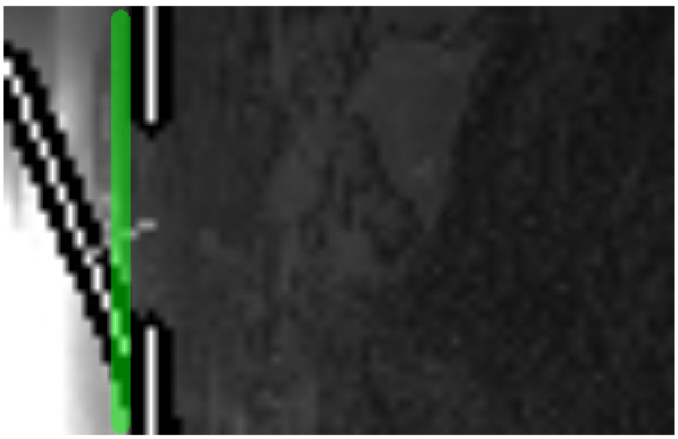

Fig. 7. Dark clouds contrasting a bright sky will mislead the horizon tracker. The estimated horizon (continuous black-white-black line) does not coincide with the real-world horizon (green line).

IV. Conclusions

This paper presents a vison-based approach to sensor FDI in gyroscopes for UAVs. A horizon tracking algorithm extracts attitude estimates from a video stream provided by a camera mounted on a fixed-wing UAV. Two Kalman filters fuse the attitude estimates with the measurements of two distinct gyroscopes for fault detection. Fault isolation is achieved by deducing the source of the fault from the innovation terms of the Kalman filters. The reuse of the on-board camera and Kalman filter state estimators allows to enhance the UAV's FDI capability without additional harware and little effort. The effectiveness of the FDI architecture was evaluated on real-time experimental data with artificially introduced sensor faults.

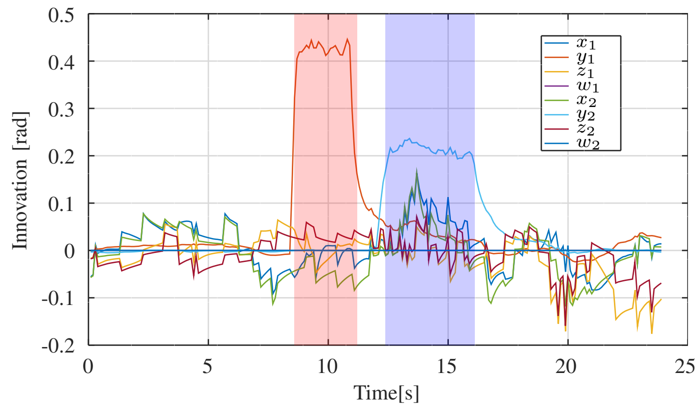

Fig. 8. Output of Kalman filter 1 and Kalman filter 2 superimposed. Measurements of GYR1 were distorted around second 10. Measurements of GYR2 were distorted around second 14. The shaded red and blue areas indicate a detected fault in GYR1 and GYR2, respectively.

References

[1] A. E. R. Shabayek, C. Demonceaux, O. Morel, and D. Fofi, "Vision based uav attitude estimation: Progress and insights," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 295–308, 2012.

[2] A. Carrio, C. Sampedro, C. Fu, J.-F. Collumeau, and P. Campoy, "A real-time supervised learning approach for sky segmentation onboard unmanned aerial vehicles," in *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on*. IEEE, 2016, pp. 8–14.

[3] K. Rudin, G. J. Ducard, and R. Y. Siegwart, "A sensor fault detection for aircraft using a single Kalman filter and hidden Markov models," in *2014 IEEE Conference on Control Applications (CCA)*. IEEE, 2014, pp. 991–996.

[4] S. Leutenegger and R. Y. Siegwart, "A low-cost and fail-safe Inertial Navigation System for airplanes," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 612–618.

[5] G. J. Ducard, *Fault-tolerant flight control and guidance systems: Practical methods for small unmanned aerial vehicles*. Springer Science & Business Media, 2009.

[6] G. Ducard and H. P. Geering, "Efficient nonlinear actuator fault detection and isolation system for unmanned aerial vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 1, pp. 225–237, 2008.

[7] S. Hansen, M. Blanke, and J. Adrian, "Diagnosis of UAV pitot tube failure using statistical change detection," in *Proceedings of the 7th IFAC Symposium on Intelligent Autonomous Vehicles, Lecce, Italy*, 2010.

[8] G. Ducard, K. Rudin, S. Omari, and R. Siegwart, "Strategies for sensor-fault compensation on UAVs: Review, discussions & additions," in *Control Conference (ECC), 2014 European*. IEEE, 2014, pp. 1963–1968.

[9] Y. Gu, J. N. Gross, M. B. Rhudy, and K. Lassak, "A Fault-Tolerant Multiple Sensor Fusion Approach Applied to UAV Attitude Estimation," *International Journal of Aerospace Engineering*, vol. 2016, 2016.

[10] P. Kryenbühl, "Sensor Fault Detection for UAVs," Bachelor's Thesis, ETH Zürich, 2016.

[11] Song, He, Hu, Shao-Lin, and Zhou, Ke-Yi, "Review on Development of Fault Diagnosis for Gyroscope," *ITM Web Conf.*, vol. 11, p. 07001, 2017. [Online]. Available: https://doi.org/10.1051/itmconf/20171107001

[12] Y. Hong Gao, D. Zhao, and Y. Bo Li, "UAV Sensor Fault Diagnosis Technology: A Survey," *Applied Mechanics and Materials*, vol. 220-223, pp. 1833–1837, 11 2012.

[13] S. M. Ettinger, M. C. Nechyba, P. G. Ifju, and M. Waszak, "Vision-guided flight stability and control for micro air vehicles," *Advanced Robotics*, vol. 17, no. 7, pp. 617–640, 2003.

[14] M. Bloesch, H. Sommer, T. Laidlow, M. Burri, G. Nuetzi, P. Fankhauser, D. Bellicoso, C. Gehring, S. Leutenegger, M. Hutter *et al.*, "A Primer on the Differential Calculus of 3D Orientations," *arXiv preprint arXiv:1606.05285*, 2016.