Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 55-14 (2022) 46–51

Optical flow-based angular rate sensor fault detection on UAVs ⋆

Peter Bauer * **Szabolcs Kun** *

* *Systems and Control Laboratory, Institute for Computer Science and Control (SZTAKI), ELKH, Budapest, Hungary (e-mail: bauer.peter@sztaki.hu)*

Abstract: The paper first extends the previous work of the authors dealing with optical flow-based angular rate estimation. Extension means consideration of camera position and orientation relative to the UAV body system (non-aligned camera). After the extension the proposed method is evaluated with virtual reality (Unreal-Carla) image sequences considering simulated aircraft flight trajectories and different frames per second (fps) rates. Based-on the results showing high fps requirements for agile flight scenarios a reversed method is proposed for angular rate sensor fault detection considering the integration of system dynamics based-on angular rate and velocity measurements and comparing the predicted image feature positions with the measured ones. The feature position differences are the error measures completed with up-down counters. The results are promising however, only basic evaluation of the proposed methods is done paving the way for detailed evaluation and development outlined in the conclusion.

Keywords: angular rate fault detection, optical flow, UAV

1. INTRODUCTION

The fault detection of aerial vehicle onboard systems is a crucial task according to Goupil et al. (2015), Heredia et al. (2005). With the widespread use of UAVs (Unmanned Aerial Vehicles) and according to the new EU regulations European Union (2019) redundancy and tolerance of single faults is mandatory also for small UAVs. Besides the application of multiple IMU and GPS sensors in case of UAVs also the on-board camera system can be considered as a possible source of motion information (Kehoe et al. (2006), Simlinger and Ducard (2019), Hiba et al. (2021)). As Song, He et al. (2017) points out the angular rate sensors are core parts of the navigation so the current work targets their fault detection. Simlinger and Ducard (2019) presents a horizon detection and so attitude based solution for gyro fault detection with Kalman filter (KF) and residual comparison. However, the horizon detection can be complicated in case of mountain areas (see Fig. 3) and it gives information only from roll and pitch motion. Their assumption of aircraft body forward axis aligned velocity vector further restricts applicability. The yaw angle seems to be included in the presented KF but its source is not dicsussed. The considered gyro faults are also not detailed only two fault scenario results are presented without in depth evaluation. So our goal here is to propose a solution which is applicable for any feature points in general 3D flight scenarios and evaluate it with a set of realistic gyro faults based-on the literature. Optical flow (OF) is widely applied for angular rate estimation in space applications for slow motions (angular rate below $10°/s$) e. g. Pal and Bhat (2013), Fasano et al. (2013). Kehoe et al. (2006) shows that it can be applied also for faster motions at least with high (30) fps camera. Contrary to Kehoe et al. (2006) nonlinear optimization-based solutions Kun and Bauer (2021) derived a two step linear solution with some approximation of the variables and compared it to the nonlinear methods considering artificial image point data.

The first goal of the current work is to extend the method from Kun and Bauer (2021) to non-body aligned and non-CG (center of gravity) placed cameras and explore its capabilities with synthetic but realistic images generated based-on simulated flight trajectories of the Sindy aircraft (SZTAKI (2014)). The second goal is to evaluate its applicability in gyro sensor fault detection. Finally, a simple OF-based angular rate fault detection method is proposed and evaluated for the typical sensor fault scenarios from Heredia et al. (2005).

Section 2 introduces the optical flow equations. Section 3 briefly describes the test flight trajectories and OF calculation for which the linear OF solution is evaluated in Section 4. Then Section 5 introduces and evaluates the proposed OF-based angular rate fault detection method. Finally, Section 6 concludes the paper.

2. OPTICAL FLOW EQUATIONS IN CAMERA COORDINATE SYSTEM

Throughout the work the considered coordinate systems are the North-East-Down (NED) earth, the X_B, Y_B, Z_B body and X_C, Y_C, Z_C camera systems the latter two shown

⋆ The research was supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Autonomous Systems National Laboratory Program. Project no. TKP2021-NVA-01 has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021-NVA funding scheme.

10.1016/j.ifacol.2022.07.581

in Fig. 1. The position of the camera in the body system is $\Delta^B = [x_b\ y_b\ z_b]^T$ and the rotation matrix from body to camera system is $T_{CB}(\phi_c, \theta_c, \psi_c)$ defined by Z-Y-X Euler angle rotations. The rotation from earth to body is defined similarly as $T_{BE}(\phi, \theta, \psi)$. To have a compact and simple form for the OF equations the body angular rate $\omega^B = [p\ q\ r]^T$ and NED velocity $V^E = [v_N\ v_E\ v_D]^T$ vectors are transformed into camera system together with the camera position vector: $\omega^C = T_{CB}\omega^B = [p_c\ q_c\ r_c]^T$, $V^C = T_{CB}T_{BE}V^E = [u_c\ v_c\ w_c]^T$ and $\Delta^C = T_{CB}\Delta^B = [x_c\ y_c\ z_c]^T$. Note that T_{CB} includes an axis swap giving r_c roll, p_c pitch and q_c yawrates if the camera system is aligned with UAV body.

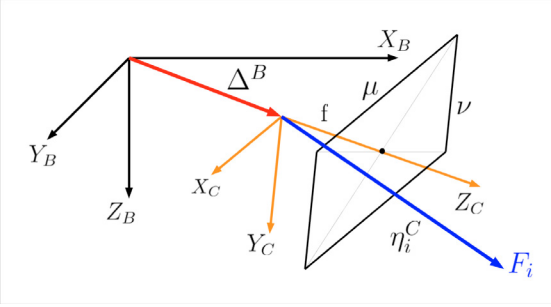

Fig. 1. The applied coordinate systems

With these notations the dynamics of the F_i (ith) feature point (corner feature) position ($\eta_i^C = [\eta_{1i}\ \eta_{2i}\ \eta_{3i}]^T$) in the camera system can be derived based-on Kehoe et al. (2006) as:

$$\dot{\eta}_i^C = \begin{bmatrix} -I_3 \end{bmatrix} \begin{bmatrix} \Delta^C \times \end{bmatrix} + \begin{bmatrix} \eta_i^C \times \end{bmatrix} \begin{bmatrix} V^C \\ \omega^C \end{bmatrix} \tag{1}$$

Here I_3 is the 3x3 unit matrix and $[V\times]$ is the matrix representation of vector cross product for any vector V. Considering the definition of image coordinates (μ_i, ν_i) of F_i with the f focal length of the camera:

$$\mu_i = f\frac{\eta_{1i}}{\eta_{3i}} \qquad \nu_i = f\frac{\eta_{2i}}{\eta_{3i}} \tag{2}$$

the image Jacobian (giving the OF) from their time derivative results as:

$$\begin{bmatrix} \dot{\mu}_i \\ \dot{\nu}_i \end{bmatrix} = \frac{f}{\eta_{3i}^2} \begin{bmatrix} \eta_{3i} & 0 & -\eta_{1i} \\ 0 & \eta_{3i} & -\eta_{2i} \end{bmatrix} \dot{\eta}_i^C \tag{3}$$

Considering $\phi_c = \psi_c = 0$ meaning only pitch angle for the camera and substituting (1) into (3) the detailed OF equations result as (considering 3D velocity and angular rate vectors):

$$\begin{aligned}
\dot{\mu}_i = &-\frac{f}{\eta_{3i}}u_c + \frac{\mu_i}{\eta_{3i}}w_c + \\
&\frac{\mu_i}{\eta_{3i}}(\eta_{2i} + \cos(\theta_c)z_c + \sin(\theta_c)x_c)p_c - \\
&\left(f + \frac{f(\cos(\theta_c)x_c - \sin(\theta_c)z_c)}{\eta_{3i}} + \frac{\mu_i^2}{f} + \frac{\mu_i y_c}{\eta_{3i}}\right)q_c + \\
&\left(\nu_i + \frac{f}{\eta_{3i}}(\cos(\theta_c)z_c + \sin(\theta_c)x_c)\right)r_c \\
\dot{\nu}_i = &-\frac{f}{\eta_{3i}}v_c + \frac{\nu_i}{\eta_{3i}}w_c + \left(f + \frac{f(\cos(\theta_c)x_c - \sin(\theta_c)z_c)}{\eta_{3i}}\right)p_c + \\
&\left(+\frac{\nu_i^2}{f} + \frac{\nu_i}{\eta_{3i}}(\cos(\theta_c)z_c + \sin(\theta_c)x_c)\right)p_c - \\
&\left(\frac{\mu_i\nu_i}{f} + \frac{\nu_i y_c}{\eta_{3i}}\right)q_c - \left(\mu_i + \frac{f y_c}{\eta_{3i}}\right)r_c
\end{aligned} \tag{4}$$

Assumption $\phi_c = \psi_c = 0$ is only considered for the sake of simplicity. As the camera angles are known from calibration the system can also be derived with complete misalignment. Considering the OF $(\dot{\mu}_i, \dot{\nu}_i)$ known for N feature points and the camera parameters $(f, \theta_c, x_c, y_c, z_c)$ also known we have $2N$ nonlinear equations for $6 + N$ unknowns $(u_c, v_c, w_c, p_c, q_c, r_c, \eta_{3i})$ for which solution techniques are presented in Kehoe et al. (2006). However, in case of on-board algorithms the computational load should be decreased as much as possible that's why a two step linear solution was proposed in Kun and Bauer (2021) assuming that the translational motion is dominated by $u_c \gg v_c$ & $u_c \gg w_c$. To be able to apply that solution for the current system (there $\Delta^C = 0$) the $x_c/\eta_{3i}, y_c/\eta_{3i}, z_c/\eta_{3i}$ terms should be evaluated as they cause η_{3i} dependence in the angular rate part of the equation also. Usually $\eta_{3i} \gg (x_c\ or\ y_c\ or\ z_c)$ can be assumed considering distant features leading to zero approximation of these terms. The resulting system is exactly the same as in Kun and Bauer (2021) so the two step linear solution can be applied. As it reduces the number of unknowns in both steps it requires at least 3 feature points but 5-6 or more is advisable.

3. TEST TRAJECTORIES AND OPTICAL FLOW CALCULATION

Test trajectories for the OF algorithm were generated with the Matlab simulation of the Sindy aircraft (SZTAKI (2014)) including a waypoint tracking controller. First, straight trajectories with different turbulence levels (small, medium, large) were generated exciting the angular rates of the system and causing different angular rates. Then a long left turn and a trajectory with multiple turns were generated to cover the full dynamic angular rate range of Sindy (see Fig. 2).

All of the trajectories were sampled with $\Delta t = 0.02s$ saving aircraft position and orientation. This data was exported to Unreal-Carla where sequences of images were generated considering $\Delta^C = [0\ 0\ 1]^T\ m$ camera position and $\theta_c = -20°$ (down) angle for a HD camera (1280 × 960 pixels with $f = 1108.5$). A city landscape surrounded by hills and vegetation was considered as the scene (see Fig. 3) and images were generated with Δt sampling (50 fps).

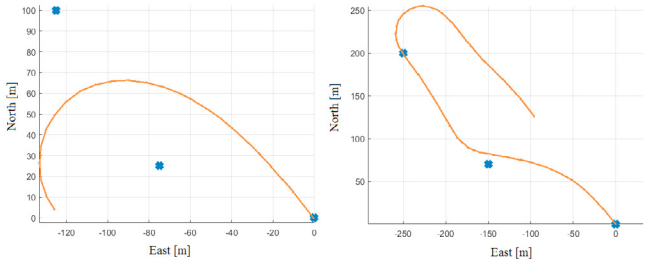

Fig. 2. Test flight trajectories (left: long turn, right: multiple turns)

Fig. 3. Example camera image from Unreal-Carla

The OpenCV functions *goodFeaturesToTrack* (Shi-Tomasi corner detector with maximum 30 corners, 0.5 quality level, 50 minimum distance and 5 blocksize) and *calcOpticalFlowPyrLK* (sparse optical flow with iterative Lucas-Kanade method applying pyramids with 15×15 window size, 3 max. level, 10 count and 0.03 epsilon) were applied to detect possible features in the first frame (index k) and track them in the second (index $k+1$). The sampling time between the frames was $\Delta T = k_s \Delta t$ considering multiple $k_s \in \mathbb{N}\backslash\{0\}$ values. The OF velocities were calculated as first order numerical derivatives:

$$\dot{\mu}_i \approx \frac{\mu_i(k+1) - \mu_i(k)}{\Delta T}, \quad \dot{\nu}_i \approx \frac{\nu_i(k+1) - \nu_i(k)}{\Delta T}$$

From the OF the angular rates of the UAV (in the camera system) were estimated with the two step linear solution from Kun and Bauer (2021) applying the approximations discussed in the previous section. Results are evaluated in the next section.

4. RESULTS OF ANGULAR RATE ESTIMATION FROM OPTICAL FLOW

First, the angular rate estimation results for the small, medium and large turbulence cases were evaluated considering $k_s \in \{10, 5, 2, 1\}$ meaning decreasing ΔT time between the images. Its important to note that mean value of the angular rates at the two image samples is plotted as the algorithm estimates the average angular rate between the two images not the instantaneous one. Results are as expected, for larger angular rates more dense sampling is required. As the roll rate was the dominant its estimation results are shown in Fig. 4 for $k_s = 10$ and in Fig. 5 for k_s values giving acceptable results (mean estimation error below $0.5°/s$).

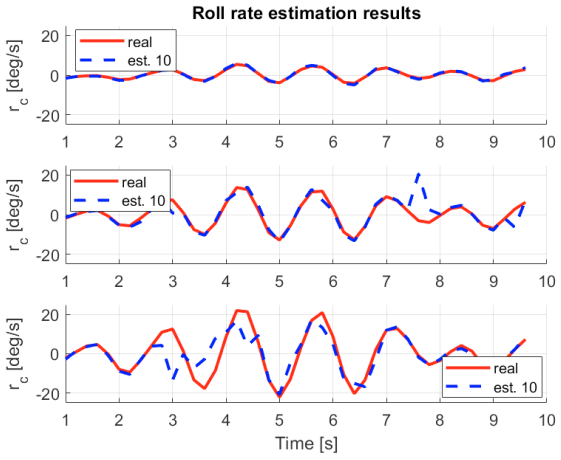

Fig. 4. Roll rate estimation results for $k_s = 10$ (small, medium, large turbulence from up to down)

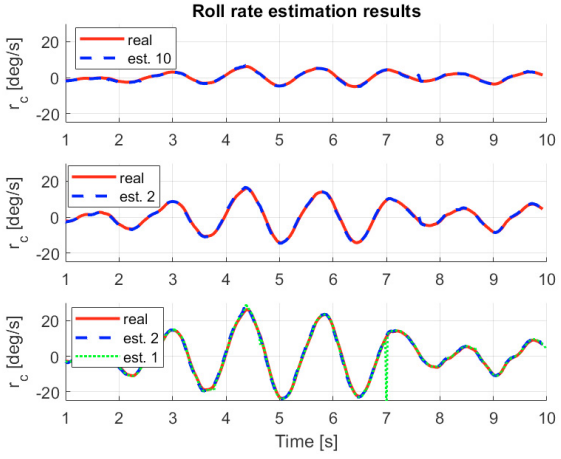

Fig. 5. Roll rate estimation results for acceptable k_s (value shown in legend) (small, medium, large turbulence from up to down)

The subfigures are plotted with the same angular rate limits to show the increase in roll rate amplitude. Fig. 4 shows that for small angular rates (maximum $5 - 6°/s$) even the sparse $\Delta T = 0.2s$ sampling provides good estimation results, however for larger rates the results are more and more unacceptable. Fig. 5 shows that for $k_s = 2 \rightarrow \Delta T = 0.04s$ (meaning 25 fps) the results are really good even for angular rates above $20°/s$. What is interesting the results get worse with $k_s = 1$ having outliers in the estimates. The cause of this is planned to be examined in the future. The results were similar for pitch and yaw rates.

In the next step estimation results for the long turn and multiple turns scenarios were evaluated having even larger

angular rate values as shown in Fig. 6. The figure shows the largest angular rate components and their estimates for $k_s = 10 \rightarrow \Delta T = 0.2s$, $k_s = 2 \rightarrow \Delta T = 0.04s$ and $k_s = 1 \rightarrow \Delta T = 0.02s$.

This case the densest sampling with $k_s = 1$ definitely gives the best results. The figure also shows that for large sampling times the OF-based angular rate estimation method is simply unable to track the larger angular rate values.

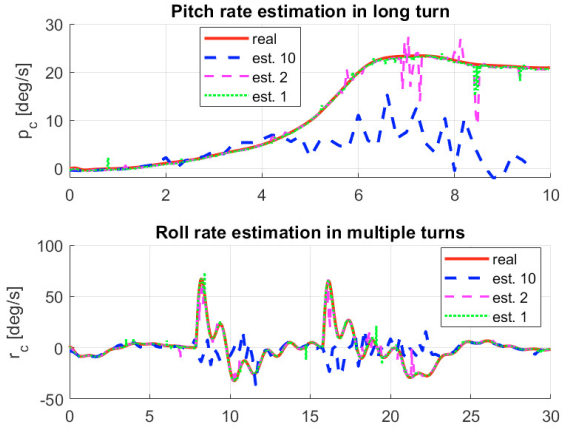

Fig. 6. Angular rate estimation results for the turning flight scenarios

The overall results show that despite neglecting the $\Delta^C = [0\ 0\ 1]^T$ camera position and assuming a dominant u_c effect the two step linear solution proposed in Kun and Bauer (2021) is able to well estimate the angular rates of the UAV with sparse sampling for moderate rates and with dense sampling for higher rates. Dense sampling means 50 fps which is not realistic with today's on-board processing hardware rather 10-12 fps are realistic (see e.g. Simlinger and Ducard (2019), Hiba et al. (2021)). The next section deals with evaluation from a fault detection point of view.

5. APPLICATION OF OPTICAL FLOW IN ANGULAR RATE SENSOR FAULT DETECTION

Results of angular rate estimation from OF in the previous section show that it is possible to accurately estimate the rates but high fps is required if the system has fast dynamics. The required fps values (25-50) are currently not realizable with small size on-board image processing hardware such as NVIDIA Jetson TX2 and Xavier NX (as preferred low cost high performance hardware) and even the best estimates have outlier data points. This latter makes comparison of estimated and measured angular rates for fault detection questionable. So possibly a low fps alternative method should be developed and used.

Assuming that the V^E velocity vector and Euler angles are available from GPS measurement and some on-board navigation algorithm V^C can be calculated at every time step. Substituting V^C and ω^C (from the measured ω^B) into (1) one has a dynamic equation for the motion of any feature point in the camera system which can be propagated in time:

$$\eta_i^C(k+\delta) = \eta_i^C(k) + \dot{\eta}_i^C(k)\Delta t \tag{5}$$

Here δ symbolizes IMU sampling step between t_k and t_{k+1} image samples having $N\delta = 1$ and so $N\Delta t = \Delta T$. Substituting (1) into (5) and recursively executing the steps one finally gets:

$$\begin{aligned} &\eta_i^C(k+1) = \prod_{j=0}^{N-1} \left(I_3 - \left[\omega^C(k+j\delta)\times\right]\Delta t\right)\eta_i^C(k) + \\ &\sum_{j=0}^{N-1}\left[\prod_{m=j+1}^{N-1}\left(I_3 - \left[\omega^C(k+m\delta)\times\right]\Delta t\right)\right]\cdot \\ &(\Delta^C \times \omega^C(k+j\delta) - V^C(k+j\delta))\Delta t = M_\eta(k)\eta_i^C(k) + V_\omega(k) \end{aligned} \tag{6}$$

Defining $\prod_{m=N}^{N-1}(.) = I_3$. Considering (2) at the two image samples and substituting (6) one gets a system of equations:

$$\begin{aligned} &\begin{bmatrix} f & 0 & -\mu_i(k) \\ 0 & f & -\nu_i(k) \end{bmatrix}\eta_i^C(k) = 0 \\ &\underbrace{\begin{bmatrix} f & 0 & -\mu_i(k+1) \\ 0 & f & -\nu_i(k+1) \end{bmatrix}}_{M_{\mu\nu}(k+1)} M_\eta(k)\eta_i^C(k) = -M_{\mu\nu}(k+1)V_\omega(k) \end{aligned} \tag{7}$$

This gives 4 equations for the 3 unknown components of $\eta_i^C(k)$ for every feature so a least squares solution is possible for any feature and theoretically only one feature is enough. Practically at least 5-6 features should be considered to have a more realistic solution. Note that the derived equations do not have any approximation, all velocity components, camera position Δ^C and orientation are considered correctly.

After obtaining the results the estimated coordinates of $\eta_i^C(k)$ can be reprojected to the image according to (2) giving $(\mu_i'(k), \nu_i'(k))$. The distances of the original $(\mu_i(k), \nu_i(k))$ and reprojected feature positions can be considered as an error measure in angular rate fault detection. After calculating the mean and standard deviation of the error vector, outliers out of the 2σ bounds are removed. The mean of the resulting set is applied as an error measure.

The test of the algorithm is done with the turning flight trajectories (see Fig. 2) assuming two IMUs with two sets of angular rate measurements (sensor 1 / 2). The possible angular rate sensor errors are well summarized in Heredia et al. (2005) and all of them is applied in the test either on roll, pitch or yaw rate of sensor 2. The errors and their considered parameters are listed below (these numbers are applied to identify errors in figures and table):

(1) Total failure (sensor gives 0 output)
(2) Stuck failure (sensor gives last value as constant output)
(3) Additive error (0.01 rad/s added to measured value)
(4) Multiplicative error (output is 90% of measured value)

The fault detection is done by comparing the error measures for sensor 1 and 2 and applying up-down counters. The counter of the sensor with larger error measure is increased, while the other is decreased in every step consid-

ering 0 lower limit for the counters. An error of the given sensor is detected when its counter is larger than the other with at least 2 value.

First the algorithm was evaluated with $k_s = 5 \rightarrow \Delta T = 0.1s$ image data but gave unsatisfactory results. Then $k_s = 4 \rightarrow \Delta T = 0.08s$ meaning 12.5 fps was applied giving satisfactory results as the table shows. 12.5 fps is an achievable goal with current state of the art on-board hardware. All the time V^C was calculated obtaining the Euler angles (with simple integration of the ideal data) from the mean values of the angular rates (from sensor 1 and 2) naturally affected by the error of sensor 2. This is to model the fact that before detection of the fault on-board attitude estimation practically considers the mean value of multiple measurement sources.

Table 1. OF-based fault detection test results

Error type	Rate	Long turn steady	Long turn varying	Multiple turns steady	Multiple turns varying
1	r_c	0.16s	0.16s	0.16s	0.16s
	p_c	0.16s	0.16s	0.24s	0.16s
	q_c	0.16s	0.16s	0.16s	0.16s
2	r_c	0.32s	0.24s	0.16s	0.32s
	p_c	0.16s	0.16s	0.16s	X
	q_c	0.16s	0.16s	0.16s	0.32s
3	r_c	0.24s	0.16s	0.16s	0.24s
	p_c	0.24s	0.4s	0.16s	X
	q_c	0.16s	0.16s	0.24s	0.16s
4	r_c	0.16s	0.16s	X	X
	p_c	0.16s	0.16s	0.24s	0.16s
	q_c	0.16s	0.16s	X	0.16s

Table 1 summarizes the test results for the different angular velocities. Steady case means that the fault is activated when the angular rates are close to constant (7s for long turn, 25s for multiple turns) while varying case means fault activation before large changes of the rates (4.5s for long turn and 8.2s for multiple turns) see Fig. 6. The table shows the detection times from fault activation in case of successful detection and X values if there is a false detection (sensor 1 is declared faulty). From the 48 scenarios 5 are false (about 10%) but from them 3 are multiplicative error cases which give good results if the output signal is 80% instead of the 90%. Possibly the 4th bias case also becomes good with larger bias. The times for successful detection are mostly 0.16s with the largest value as 0.4s which are all acceptable. Upon evaluation of the results it should be noted that ideal angular rate and GPS velocity values were considered from Matlab simulation without any noise component or calibrated sensor bias. That's why there was no tolerance range between the error measures upon decision about their ratio. In more realistic cases a threshold should be applied decreasing the chance of false detection.

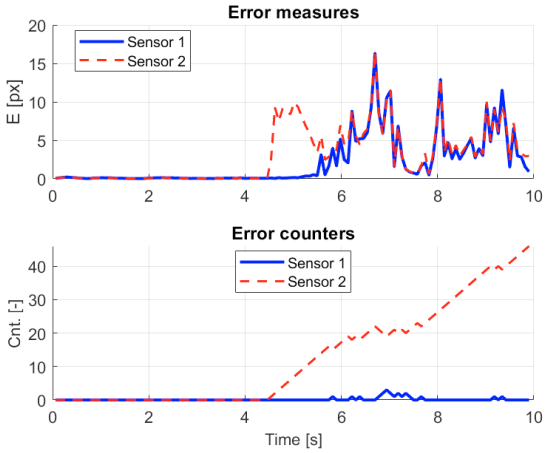

Fig. 7. Good estimation results for long turn with varying rate, type 1 fault on roll rate

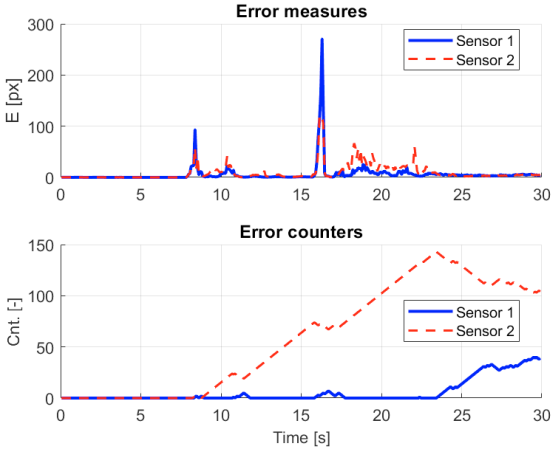

Fig. 8. Satisfactory estimation results for multiple turns with varying rate, type 2 fault on pitch rate

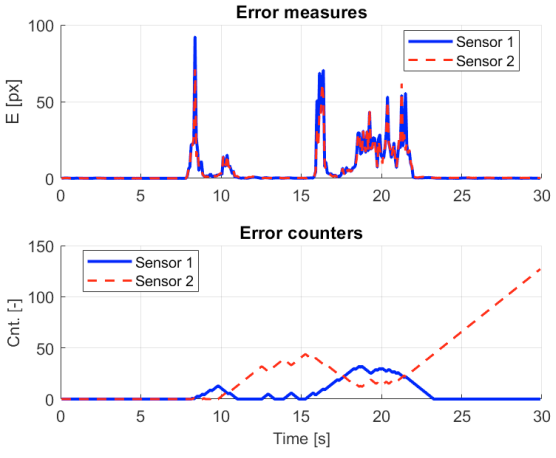

Fig. 9. Unsatisfactory estimation results for multiple turns with varying rate, type 3 fault on pitch rate

Fig.s 7 to 9 show error measures and error counters in a good, a satisfactory and an unsatisfactory case, the former

two giving good decision while the latter a false detection. The first two figures well show that most of the time the error measure of the faulty sensor is larger giving a good decision with the application of the counters. Note that larger can mean only a slight difference as in Fig. 7 after 6 seconds. In the future for real flight data some thresholding will be applied avoiding counter changes because of slight differences. In the last case there are two sections (the high angular rate ones) when the error measure of the good sensor is higher giving a false decision. This is possibly caused by a large number of falsely tracked features by the Lucas-Kanade algorithm giving inconsistent image data.

6. CONCLUSION

The paper first extends a previously developed method of the authors for the estimation of angular rates from on-board camera optical flow. Extension means consideration of camera position and orientation relative to the UAV body coordinate system. The method is then evaluated on a series of images generated considering realistic simulated trajectories of a UAV in Unreal-Carla virtual environment. Straight flight with turbulence and flights with turns are considered. The results are promising but in case of high angular rates 25-50 *fps* image stream is required which is unrealistic in case of on-board image processing with reasonable cost technology.

That's why for angular rate fault detection a reverse approach is proposed integrating the measured UAV angular rates and velocities through the optical flow equations to estimate image feature motions between two frames. Comparison of estimated feature positions with the measured ones (considering the average feature distances as error measure) makes it possible to detect faults in the angular rates. An advantageous property of this approach can be that the systematic errors are included for all sensors and so the faulty sensor should give a larger difference irrespective of the systematic error. The method is evaluated considering two angular rate sensors one fault-free and one faulty in one direction. All of the possible angular rate errors according to the literature are considered. The results are promising considering only 12.5 *fps* image frequency but detailed evaluation and fine tuning are required.

Future research should include the evaluation of the applied Lucas-Kanade optical flow calculation trying to remove falsely tracked features (in this article all tracked features were considered correct), the evaluation of the optical flow-based angular rate calculation for high cross-wind scenarios (UAV motion possibly not dominated by forward velocity), the consideration of angular rate and velocity measurement noises and real flight data, the possible reformulation considering inter-frame attitude changes instead of the angular rates and the evaluation of the possibility of velocity error detection. Finally, real flight tests are planned to validate the developed approaches.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the help of Antal Hiba (Computational Optical Sensing and Processing Laboratory, SZTAKI) with the generation of the Unreal-Carla image series based-on our Sindy simulated flight trajectories.

REFERENCES

European Union (2019). COMMISSION DELEGATED REGULATION (EU) 2019/945 of 12 March 2019 on unmanned aircraft systems and on third-country operators of unmanned aircraft systems.

Fasano, G., Rufino, G., Accardo, D., and Grassi, M. (2013). Satellite Angular Velocity Estimation Based on Star Images and Optical Flow Techniques. *Sensors*, 13(10), 12771–12793. doi:10.3390/s131012771. URL `https://www.mdpi.com/1424-8220/13/10/12771`.

Goupil, P., Boada-Bauxell, J., Marcos, A., Rosa, P., Kerr, M., and Dalbies, L. (2015). An overview of the FP7 RECONFIGURE project: industrial, scientific and technological objectives. In *International Federation of Automatic Control (IFAC)*, 976–981.

Heredia, G., Ollero, A., Mahtani, R., Bejar, M., Remuss, V., and Musial, M. (2005). Detection of Sensor Faults in Autonomous Helicopters. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2229–2234. doi:10.1109/ROBOT.2005.1570444.

Hiba, A., Gati, A., and Manecy, A. (2021). Optical Navigation Sensor for Runway Relative Positioning of Aircraft during Final Approach. *Sensors*, 21(6). doi:10.3390/s21062203. URL `https://www.mdpi.com/1424-8220/21/6/2203`.

Kehoe, J., Watkins, A., Causey, R., and Lind, R. (2006). State Estimation Using Optical Flow from Parallax-Weighted Feature Tracking. In *In Proc. of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics. doi:10.2514/6.2006-6721. URL `https://doi.org/10.2514/6.2006-6721`.

Kun, S. and Bauer, P. (2021). Improvements in Optical Flow-based Aircraft Partial State Estimation. In *2021 17th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA)*, 1–4. doi:10.1109/CNNA49188.2021.9610822.

Pal, M. and Bhat, M.S. (2013). Star sensor based spacecraft angular rate estimation independent of attitude determination. In *2013 IEEE International Conference on Control Applications (CCA)*, 580–585. doi:10.1109/CCA.2013.6662812.

Simlinger, B. and Ducard, G. (2019). Vision-based Gyroscope Fault Detection for UAVs. In *2019 IEEE Sensors Applications Symposium (SAS)*, 1–6. doi:10.1109/SAS.2019.8705965.

Song, He, Hu, Shao-Lin, and Zhou, Ke-Yi (2017). Review on Development of Fault Diagnosis for Gyroscope. *ITM Web Conf.*, 11, 07001. doi:10.1051/itmconf/20171107001. URL `https://doi.org/10.1051/itmconf/20171107001`.

SZTAKI, M. (2014). Sindy test aircraft. URL `http://uav.sztaki.hu/sindy/home.html`.