Joint Virtual and Physical Prototype Design and Testing of a Sensor Fusion Workbench for Fixed-Wing UAVs

Peng Huang[1], Heinrich Meyr[2], and Gerhard Fettweis[1,2]

***Abstract*— Unmanned aerial vehicles (UAVs) require cost-efficient on board multi-sensor fusion to achieve accurate and reliable flight state estimation. The challenges behind the implementation of sensor fusion algorithms from scratch towards in-flight testing on microcontroller-based hardware are, however, demanding, since it requires not only understanding and implementing complex sensor fusion algorithms but also developing embedded software in a microcontroller. Those challenges make the process of prototyping onboard sensor fusion algorithms time-consuming, expensive, and inefficient. We present fast prototyping of a sensor fusion workbench based on extended Kalman filtering (EKF) for fixed-wing UAVs. The workbench incorporates multiple sensors, including an inertial measurement unit, a GPS receiver, and static and dynamic pressure sensors. The multi-sensor fusion algorithm has been tested under virtual flight data, as well as measured flight data including challenging environments with thermals and gusts. After performance evaluation, we transferred the algorithm from MATLAB code to C code and integrated it into an avionics device LXNAV S10. The implemented sensor fusion algorithm has been successfully tested on a manned glider under windy and turbulent environment.**

I. Introduction

Multiple sensors have been extensively used on unmanned aerial vehicles (UAVs) due to their small size, low cost, and light weight. However, the different measurements are noisy. UAVs require cost-efficient on board multi-sensor fusion to achieve highly accurate and reliable flight state estimation. The challenges behind the implementation of sensor fusion algorithms from scratch to being applied in hardware are, however, demanding, since it requires not only understanding and implementing complex sensor fusion algorithms but also developing embedded software in a microcontroller. Those challenges make the process of prototyping on board sensor fusion algorithms time-consuming, expensive, and inefficient.

A sensor fusion workbench employs a highly sophisticated nonlinear optimal state estimation algorithm and a large number of sensors. Extended Kalman filtering (EKF) has shown to be an effective sensor fusion technique for real-time on board applications [1], [2], [3]. The mathematical theory of the algorithms has been developed over many years. Due to high cost, the application of the algorithms has been for many years restricted to space and military applications. Due to the enormous progress in semiconductor technology today, sensors are cheap, and a tremendous amount of computing power in signal processors is available at low cost. This makes the use of the EKF feasible at low cost for UAV and glider applications.

[1]Vodafone Chair Mobile Communications Systems, Technische Universität Dresden, 01062 Dresden, Germany, {peng.huang, gerhard.fettweis}@tu-dresden.de

[2]Barkhausen Institut, 01062 Dresden, Germany, meyr@iss.rwth-aachen.de

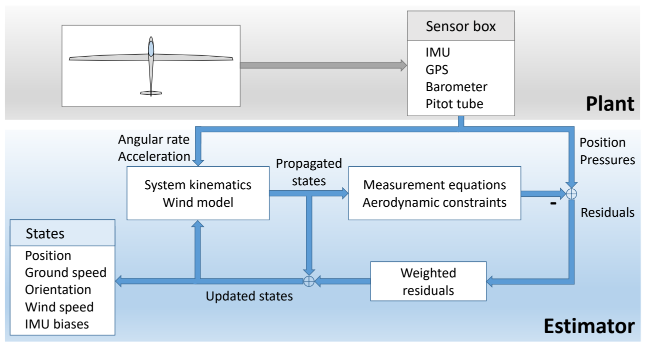


Fig. 1. Sensor fusion workbench based on an extended Kalman filter for inertial navigation and wind estimation.

Virtual prototyping is an effective way to reduce the cost and duration of developing sensor fusion algorithms for an embedded system. Furthermore, it allows a first feasibility study of the performance of the algorithms. We present fast prototyping of a sensor fusion workbench based on a nonlinear extended Kalman filter for fixed-wing UAVs. The basic idea of the state estimator known in the literature as an extended Kalman filter [1], [4], [5], [6] is shown in Fig. 1. The workbench in the avionics of a glider runs a mathematical model of the fixed-wing UAV and its dynamics in real-time. It comprises a sensor box with an inertial measurement unit (IMU), static and dynamic pressure sensors, as well as GPS. The EKF estimates the position, the ground speed, the orientation, the wind speed, and the IMU biases. We have implemented a virtual prototype of the EKF in MATLAB, which allows a comprehensive analysis of the nonlinear and time-varying EKF. On the workbench, we have used virtual and recorded flight data to validate the sensor fusion based estimator. The virtual flight data are generated in an open-source flight simulator, which provides ground truth and enables any desired flight maneuvers and various environmental conditions. The recorded flight data are provided by our industrial partner LXNAV through their commercial avionics device S10 onboard a manned glider. Pilot-in-the-loop flight allows high-quality data acquisition in different flight maneuvers, such as soaring in thermals, stall flight, and free-fall flight.

The simulation in MATLAB shows that the nonlinear EKF has excellent tracking and good acquisition behavior for both virtual and real recorded flight data. The EKF estimates the 3D air mass movement, and it does not show false signals of the total energy compensation variometer [7]. We demonstrate this by comparing the two methods using real data in turbulent air. Beyond the states, the sensor fusion framework can also provide other vital variables of the aircraft, such as the angle of attack (AoA) and the sideslip angle (SSA). Furthermore, the workbench enables us to get an in-depth insight into complex sensor fusion algorithms. For example, we have experimentally verified the theoretical observability conditions published in [8]. Last but not least, we transferred the sensor fusion algorithm from MATLAB code to C code and integrated the algorithm in the S10 from LXNAV. Flight tests showed that the sensor fusion algorithm works as predicted in theory.

The paper is organized as follows. In Section II, we briefly describe the considered flight state estimator based on an EKF. In Section III and Section IV we experimentally validated the developed algorithm. Section V describes the procedure to get from the virtual prototype of the sensor fusion algorithm in MATLAB to embedded software in an avionics device. Finally, Section VI gives a conclusion and discusses some future work.

Notation: Vectors and matrices are indicated by bold font. The superscript T denotes a transpose. Measurements are denoted by a tilde, e.g., $\tilde{p}_d$, and estimates are denoted by a hat, e.g., $\hat{\boldsymbol{r}}$. Dots denote time derivatives, e.g., $\dot{\boldsymbol{r}}$. The subscripts N and B denote the north-east-down (NED) and the body coordinate, respectively. We represent the orientation by a quaternion $\boldsymbol{q}$ and $\boldsymbol{q}_{NB}$ denotes the orientation of the body coordinate with respect to the NED coordinate. We use quaternion multiplication denoted by $\otimes$.

II. Flight State Estimator

Fig. 1 shows the flight state estimator based on the EKF. The estimator is developed based on the work of Leutenegger [1], [5]. The state vector of the underlying system contains the 3D position $\boldsymbol{r}$ (including latitude ϕ, longitude λ and height h), the 3D ground speed ${}_N\boldsymbol{v}$ in the NED coordinate, the 4D quaternion $\boldsymbol{q}_{NB}$, the 3D wind speed ${}_N\boldsymbol{d}$, the 3D gyro bias $\boldsymbol{b}_g$, and the 3D accelerometer bias $\boldsymbol{b}_a$:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{r}^{\mathrm{T}} & {}_N\boldsymbol{v}^{\mathrm{T}} & \boldsymbol{q}_{NB}^{\mathrm{T}} & {}_N\boldsymbol{d}^{\mathrm{T}} & \boldsymbol{b}_g^{\mathrm{T}} & \boldsymbol{b}_a^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{19}. \quad (1)$$

The system dynamics are described by the kinematics of the UAV driven by the acceleration ${}_B\tilde{\boldsymbol{a}}$ and the angular rate ${}_B\tilde{\boldsymbol{\omega}}$, and the assumptions on the wind speed and the biases from the IMU. In summary, the time-derivatives of the states $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{w})$ are described as [1], [5], [6]

$$\dot{\boldsymbol{r}} = \mathrm{diag}\begin{bmatrix} \frac{1}{R_\phi + h} & \frac{1}{(R_\lambda + h)\cos(\phi)} & -1 \end{bmatrix} {}_N\boldsymbol{v}, \quad (2)$$

$${}_N\dot{\boldsymbol{v}} = \boldsymbol{C}_{NB}({}_B\tilde{\boldsymbol{a}} - \boldsymbol{b}_a - \boldsymbol{w}_a) + {}_N\boldsymbol{g}, \quad (3)$$

$$\dot{\boldsymbol{q}}_{NB} = \frac{1}{2}\boldsymbol{q}_{NB} \otimes ({}_B\tilde{\boldsymbol{\omega}} - \boldsymbol{b}_g - \boldsymbol{w}_g), \quad (4)$$

$${}_N\dot{\boldsymbol{d}} = \boldsymbol{w}_d, \quad (5)$$

$$\dot{\boldsymbol{b}}_g = \boldsymbol{w}_{b_g}, \quad (6)$$

$$\dot{\boldsymbol{b}}_a = -\frac{1}{\tau}\boldsymbol{b}_a + \boldsymbol{w}_{b_a}, \quad (7)$$

with the white Gaussian processing noises which are mutually independent:

$$\boldsymbol{w} = \begin{bmatrix} \boldsymbol{w}_a^{\mathrm{T}} & \boldsymbol{w}_g^{\mathrm{T}} & \boldsymbol{w}_d^{\mathrm{T}} & \boldsymbol{w}_{b_g}^{\mathrm{T}} & \boldsymbol{w}_{b_a}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_c). \quad (8)$$

Moreover, ${}_N\boldsymbol{g} = \begin{bmatrix} 0 & 0 & 9.81\mathrm{m/s^2} \end{bmatrix}^{\mathrm{T}}$ denotes the gravity field in NED frame. And $\boldsymbol{C}_{NB}$ denotes the coordinate transformation matrix from the body to the NED coordinate. The local radii of the Earth are denoted by R_ϕ and R_λ [9]. The acceleration bias is modeled as a first-order random walk with a time constant τ.

A. Extended Kalman Filter

Based on the time-continuous state equations (2)-(7), a time-discrete extended Kalman filter is derived. In a first step, a time-discrete model is obtained by an Euler forward integration over a small time interval. In a second step, the nonlinear discrete-time state equations are linearized around a nominal trajectory. For the small error signal, an error state (extended) Kalman filter is derived based on [4], [6]. For a detailed derivation of the extended Kalman filter for the present application see [5].

We briefly describe the sensors used in EKF. The signals of the 3D axis accelerometer and of the 3D axis rate gyro are processed directly in the propagation step of the EKF. The measurement update corrects the a priori states based on the difference between the measurements (or constraints) $\tilde{\boldsymbol{y}}$ and the values calculated by the measurement model $\boldsymbol{h}(\boldsymbol{x})$ based on the a priori states.

$$\tilde{\boldsymbol{y}} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{\nu}, \quad (9)$$

with the white Gaussian measurement noise $\boldsymbol{\nu} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$. The measurement model comprises the position, the static pressure p_s, the dynamic pressure p_d, and aerodynamic constraints including the side force Y and sink rate v_s:

$$\tilde{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{r}^{\mathrm{T}} & p_s & p_d & Y & v_s \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^7. \quad (10)$$

Here the static pressure changes at different heights [9], and the dynamic pressure results from the airspeed due to the airflow [10]. Furthermore, two aerodynamic constraints of fixed-wing airplanes are used. The first constraint is the side force Y due to the SSA [5], [10, p. 60].

$$Y = p_d S C_Y, \quad (11)$$

with S being the wing area and C_Y being the side force coefficient approximately proportional to the SSA β [10, p. 61]. The second constraint characterizes the sink rate v_s as a function of the forward true airspeed (TAS) [5]

$$v_s = a_{2\,B}v_{tx}^2 + a_{1\,B}v_{tx} + a_0, \quad (12)$$

with a_0, a_1 and a_2 being constants of the speed polar, and ${}_Bv_{tx}$ being the x-component of the true airspeed in the body frame, i.e., the component along the longitudinal axis of the UAV.

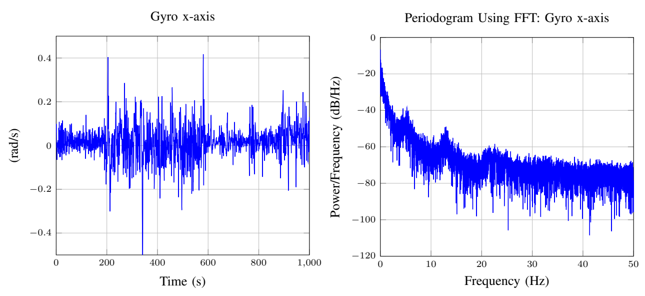


Fig. 2. Angular rate: a) measurement by gyro b) periodogram using FFT.

TABLE I
SENSOR CHARACTERISTICS

	Value	Unit
Gyro noise density	3.0e-4	$\mathrm{rad}/(\mathrm{s}\sqrt{\mathrm{Hz}})$
Gyro bias noise density	1.0e-5	$\mathrm{rad}/(\mathrm{s}^2\sqrt{\mathrm{Hz}})$
Accel. noise density	1.0e-2	$\mathrm{m}/(\mathrm{s}^2\sqrt{\mathrm{Hz}})$
Accel. bias noise density	1.0e-4	$\mathrm{m}/(\mathrm{s}^3\sqrt{\mathrm{Hz}})$
GPS position noise (NED) STD	2.2, 2.2, 10	m
Static pressure noise STD	9.5	Pa
Dynamic pressure noise STD	5.3	Pa
IMU rate	100	Hz
GPS rate	10	Hz
Pressure sensors rate	50	Hz

B. Measurement Processing

The measurements are considered to be corrupted by additive white Gaussian noises. Sensor manufacturers provide sensor specifications. Alternatively, the noise standard deviation of individual sensors can be estimated experimentally based on a periodogram of a time-series of measurements. For example, Fig. 2 shows $1000\,\mathrm{s}$ data measured by a gyro and its periodogram calculated by a fast Fourier transform (FFT). From the periodogram, the noise standard deviation of the angular rate can be determined as $3.0\times10^{-3}\,\mathrm{rad/s}$. The power spectral density of the additive noise is $9.0\times10^{-8}\,\mathrm{rad}^2/(\mathrm{s}^2\mathrm{Hz})$. The gyro data are measured at $100\,\mathrm{Hz}$. The other sensor noise characteristics can be determined in the same manner. Apart from the periodogram, the sensor noises can also be characterized by the Allan variance [11]. Table I gives the statistics of the accelerometer, the gyro, the GPS, the barometer, and the Pitot tube, in which the noise characteristics are determined through the periodogram analysis.

C. State Initialization

The position and velocity can be initialized by measurements from the GPS receiver. Since the sensors cannot directly provide the vehicle's orientation, proper initialization of orientation is of crucial importance. The orientation in terms of Euler angles can be computed from the acceleration measurements

$$\mathrm{roll} = \arctan_2({}_B\tilde{a}_y, {}_B\tilde{a}_z), \tag{13}$$

$$\mathrm{pitch} = \arctan_2\left({}_B\tilde{a}_x, \sqrt{{}_B\tilde{a}_y^2 + {}_B\tilde{a}_z^2}\right), \tag{14}$$

TABLE II
TECHNICAL DATA OF GLIDER ASG 29.

	Value	Unit
Wing area	10.5	m^2
Wing span	18	m
Mass	360	kg
Side force coeff. w.r.t β	-1.1	rad^{-1}

TABLE III
MEAN AND STANDARD DEVIATION OF ESTIMATION ERRORS.

	Mean	Standard Deviation
Position (m)	[-0.35, 0.026, 0.26]	[4.88, 3.18, 3.82]
Ground speed (m/s)	[-0.045, 0.032, 0.12]	[1.69, 1.61, 0.90]
Euler angles (deg)	[-0.042, -1.12, 0.071]	[0.99, 1.50, 2.89]
Wind speed (m/s)	[-0.0073, 0.16, -0.032]	[0.63, 0.51, 0.65]
Angle of attack (deg)	-0.4726	1.6509
Sideslip angle (deg)	0.0854	0.8255

where $\arctan_2$ denotes a four-quadrant arctangent function, which returns values in the closed interval [-π, π]. We initialize the yaw angle by the ground course from the GPS since the yaw is assumed to be close to the ground course when there is no strong crosswind. If the sensor fusion algorithm is activated before the aircraft takes off, this assumption is valid. The Euler angles are converted to the quaternion. The wind speed, the accelerometer bias, and the gyro bias are initialized by zeros.

D. Sampling Rates of Sensors

Multiple sensors have different sampling rates. Due to the asynchronous measurements, the propagation update and the measurement update are performed at different rates. Since the state equations are driven by the input of the IMU measurements, the propagation update is running at the rate of the IMU, which usually has a higher rate than the GPS and the pressure sensors. The measurement update is carried out in case that new GPS measurements and pressures are captured. E.g., the update of the GPS position occurs at the rate of $10\,\mathrm{Hz}$, and the update based on the aerodynamic constraints and pressures are sampled at the rate of the pressure sensors, i.e., $50\,\mathrm{Hz}$.

III. VALIDATION WITH VIRTUAL FLIGHT DATA

This section aims to evaluate the performance of the sensor fusion algorithm under virtual flight data, which are generated in an open-source flight simulator, FlightGear. The advantages of using FlightGear are that the generated records provide ground truth for validation of the considered algorithm. More importantly, any desired flight maneuvers and environmental conditions can be generated by FlightGear. Moreover, the simulator provides a variety of gliders and airplanes accompanied by detailed aerodynamic models. Here we use a glider ASG 29 for the evaluation of the performance of the estimator. The ASG 29 is also used in the later real flight tests. Table II lists the technical data of the glider ASG 29.

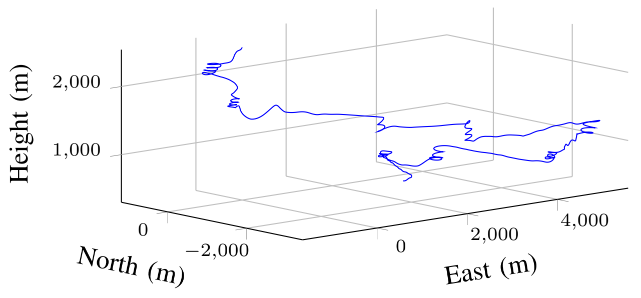


Fig. 3. Virtual flight trajectory in 3D view.

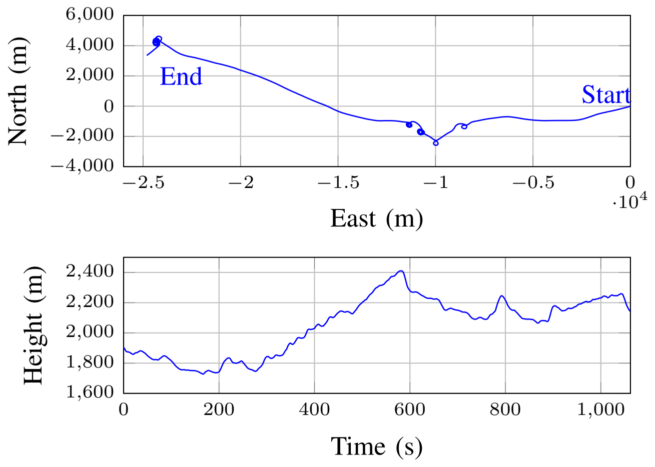


Fig. 4. Path and height estimates based on real recorded flight data.

A set of flight data was collected with the virtual ASG 29 in the flight simulator. The shown flight lasts about 1000s and contains cruising and turning maneuvers, as Fig. 3 depicts. All measurements are calculated based on the collected flight data with additive Gaussian noises as specified in Table I. The wind in the FlightGear simulation is assumed constant. Due to the availability of the ground truth, the estimates can be compared with the true values from FlightGear. Table III provides means and standard deviations of the estimation errors. The results show that the estimator has a good tracking performance not only for the states but also for the AoA and the SSA.

IV. Evaluation using Real Measurements

The implemented sensor fusion algorithm has to be evaluated based on real flight measurements in which realistic measurement noises and environments like random wind speed or gusts are considered. In a series of test flights with the gliders ASG 29 and ASH 25M, a large number of flight records have been collected by an avionics device, LXNAV S10. Table I lists the sensor specification.

A. Results of State Estimation

The flight records were collected during typical weather conditions ranging from smooth air to very turbulent and gusty wind conditions. A typical trajectory is shown in Fig. 4. Fig. 5 and Fig. 6 show the orientation estimates in

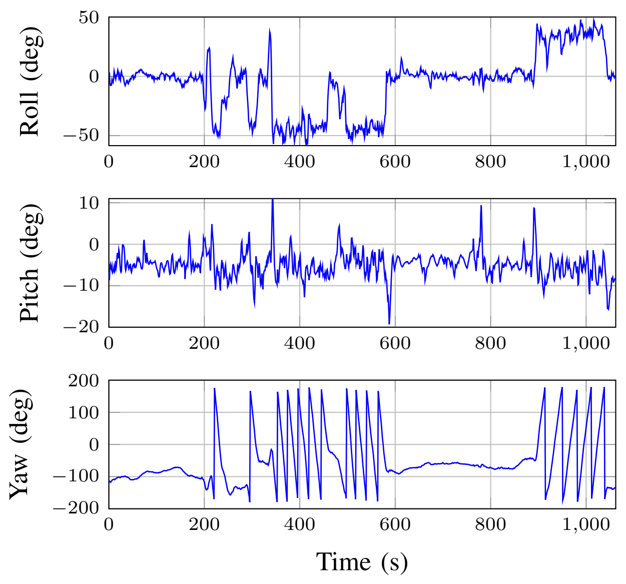


Fig. 5. Orientation estimates.

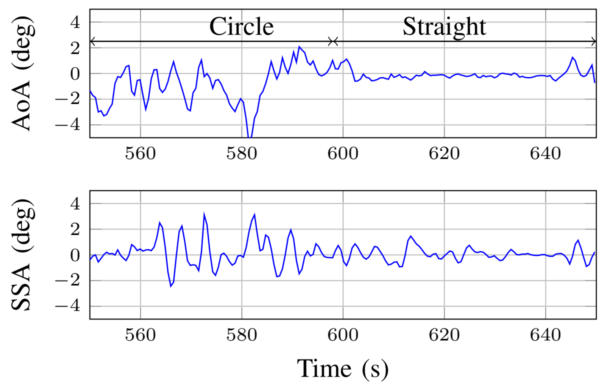


Fig. 6. Angle of attack and sideslip anlge estimates.

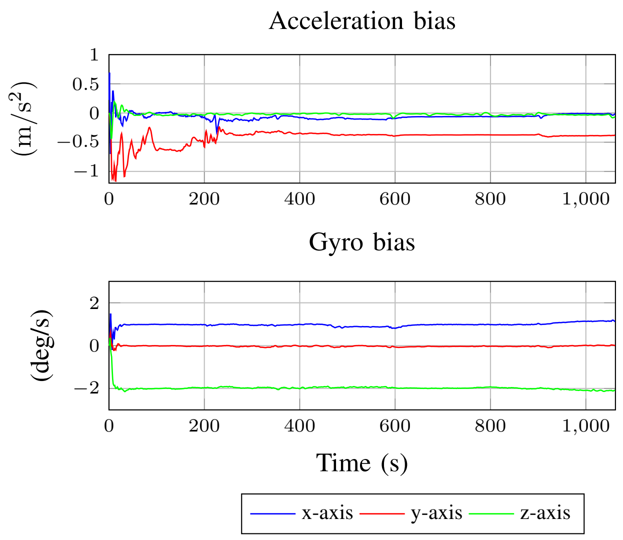


Fig. 7. IMU bias estimates.

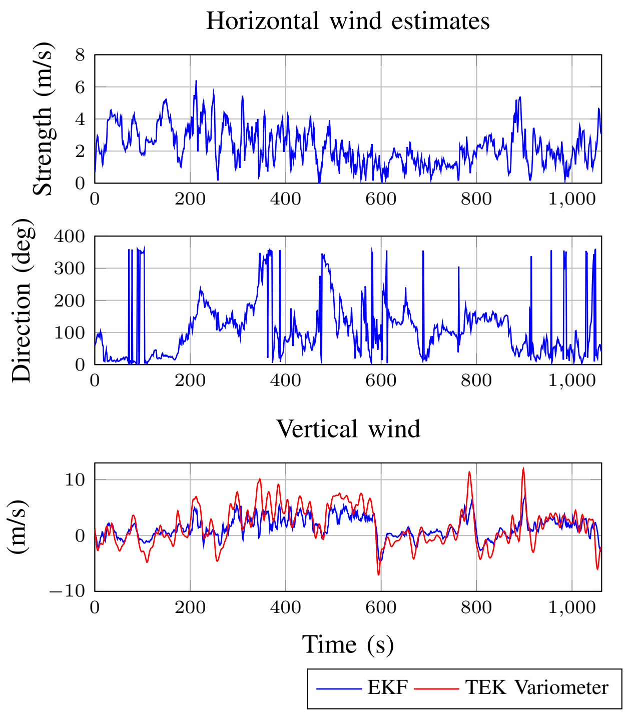


Fig. 8. Wind estimates (wind processing noise density $0.05\,\mathrm{m}/(\mathrm{s}^2\sqrt{\mathrm{Hz}})$).

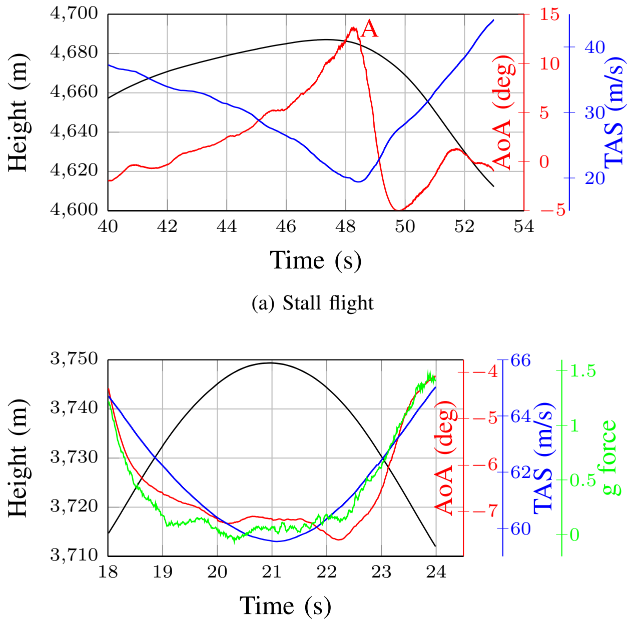
terms of Euler angles, the AoA, and the SSA. Fig. 7 shows that the estimator can cope with the biases of the low-cost accelerometer and gyro. Fig. 8 shows wind estimates which depends on the wind random walk model assumed in the EKF, see (5). The noise power spectral density of $\boldsymbol{w}_d$ is a parameter characterizing the assumed variability of the wind. The choice of this parameter is a trade-off between rapid changing estimates or filtering out fluctuations due to noise.

B. Angle of Attack Estimation

We have evaluated the estimation of the angle of attack (AoA), which is a crucial parameter for the safe operation of fixed-wing planes. The AoA can be increased up to the stall AoA, after which the lift suddenly decreases [10]. While in a commercial plane, expensive sensors for the measurement of the AoA are used, we estimate the AoA indirectly by sensor fusion via the EKF. For a stall flight and a parabolic flight, the estimate of the AoA has been plotted in Fig. 9. The stall flight in Fig. 9a shows that the critical AoA (14°) occurs at point A (48 s) when the TAS reaches the minimum speed. On the other hand, a free fall flight can be used to evaluate the minimum AoA of the plane, see Fig. 9b (−7° for the glider ASG 29).

C. Evaluation of Vertical Wind Estimation

The total energy compensation (TEK) variometer is used today in all modern gliders to estimate the vertical air mass movement [7], [12], [13]. The TEK variometer is based on the principle that in calm air the total energy of the glider is constant. A change in kinetic energy, e.g., pulling the stick, results in a change of the potential energy of exactly the same amount. In the TEK, it is assumed that the horizontal kinetic energy remains approximately constant; otherwise, the TEK variometer produces false estimates for the vertical air mass movements. We observe in Fig. 10 that the vertical wind from the TEK is dependent on the change of the horizontal TAS (acceleration). For a well compensated TEK variometer, a negative acceleration (pulling up) produces a false positive vertical wind speed, whereas pushing the stick results in a false negative vertical wind speed. In the presented case, the variometer is poorly calibrated. The difference between the vertical wind estimates of the EKF and the variometer is therefore only qualitative. Since the EKF estimates the 3D air mass movement, these false vertical wind readings do not occur, as shown in Fig. 10.



(a) Stall flight

(b) Free fall flight

Fig. 9. Estimates in (a) a stall flight and (b) a free fall flight.

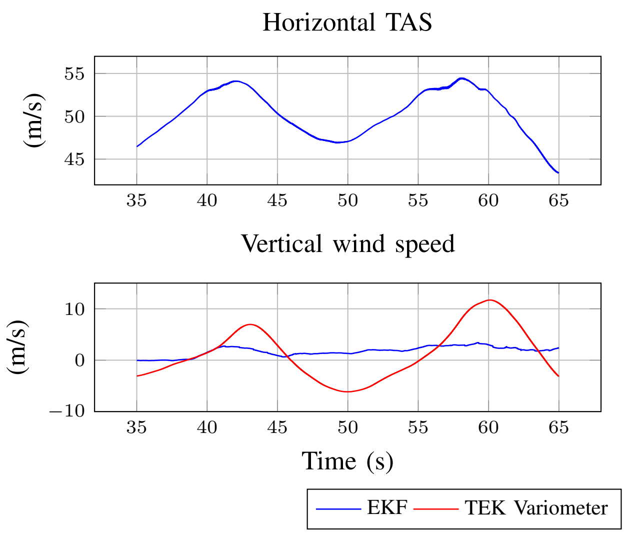


Fig. 10. Comparison between vertical wind estimates from the EKF and the TEK variometer.

V. From Virtual Prototyping to Flight Testing

This section presents the process from the virtual workbench in MATLAB towards hardware-in-the-loop (HIL), as Fig. 11 shows. Before flight testing, we can use recorded flight measurements to validate results from the algorithm in MATLAB and the avionics device S10. In Sec. III and Sec. IV, we have evaluated the developed sensor fusion algorithm in MATLAB based on virtual and real flight measurements. In the workbench in MATLAB, we can conduct observability and stability analysis to get insights into the EKF based fusion algorithm. Moreover, the workbench is accompanied by a visualization tool developed in a MATLAB GUI, which allows the visualization of sensor data and estimation results.

To target the HIL, we have to transfer the sensor fusion algorithm from MATLAB code to C code. To get an early prototype in C code, we use the MATLAB Coder™ to do the code transformation. This approach allows a quick compilation of the algorithm in a hardware application without potential coding errors caused by hand-coding. However, while the MATLAB Coder™ in the present application is well suited for prototype implementation, it does not deliver product quality code. The EKF has 19 states and employs intensive matrix computations. The matrix computations are unreadable in the generated C code since the MATLAB Coder™ expands matrix calculation by loop unwinding. Due to the loop unwinding, the code execution is also very time-consuming, hence energy-inefficient. Therefore it is not suited for use in real-time and energy-critical applications (such as small drones) running on a microprocessor with limited processing power. For ease of maintenance, high execution and energy efficiency we have written the C code using a library for linear algebra: *Eigen* [14].

As shown in Fig. 11, the C code is tested against the reference code in MATLAB in two steps to validate if they produce the same result. In the first step, the MATLAB code is tested against the C code implementation. Then it is tested against the implementation on the LXNAV S10 avionics hardware built on an ARM Cortex-M4 processor running at up to $160\,\mathrm{MHz}$. In this step, the priority is to guarantee the real-time execution of the algorithm. We experimentally measured the maximum execution time of a complete EKF iteration including the propagation and measurement update. One complete EKF iteration takes maximal $7\,\mathrm{ms}$, which is less than the $10\,\mathrm{ms}$ of the IMU sampling interval. Therefore, the real-time requirement of the algorithm is satisfied. Finally, the complete sensor fusion algorithm is tested in a series of flights with the manned gliders ASG 29 and ASH 25M. The results of these tests were recorded and used to analyze and improve the performance of the sensor fusion algorithm.

The HIL methodology presented has shown to be very efficient to shorten the development cycle and making efficient use of scarce engineering resources at separate institutions.

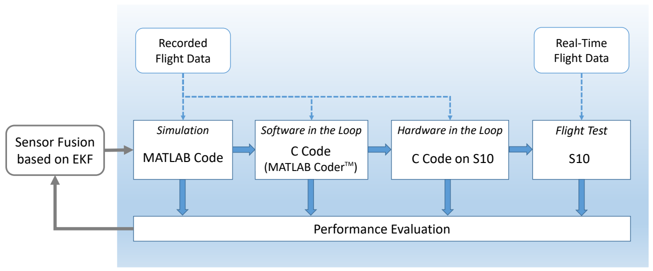


Fig. 11. Process from virtual prototyping to flight testing.

VI. Conclusion and Outlook

The framework developed enables fast transformation from the MATLAB virtual prototype to the product quality C-implementation of the sensor fusion workbench on the LXNAV S10 avionics device. The closed-loop development platform, including test flights, allows rapid iteration of the development cycle based on the results of the test flights.

We have used this platform for prototyping and experimental validation of the EKF based sensor fusion workbench for UAVs and gliders. Using virtual and recorded sensor data, we have validated that the sensor fusion workbench has an excellent performance. The estimator provides accurate flight state estimation for the position, ground speed, orientation, and 3D wind for fixed-wing airplanes. It also provides estimates of the AoA and SSA without additional sensors.

Acknowledgment

We are very grateful to Erazem Polutnik and Marko Jelenko of LXNAV for their support with integration of our algorithm on the S10, for providing recorded flight measurements, and for flight tests. We would like to thank Dr. Meik Dörpinghaus for reading our manuscript and for providing helpful suggestions. This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant FE 423/20-1.

References

[1] S. Leutenegger, A. Melzer, K. Alexis, and R. Siegwart, "Robust state estimation for small unmanned airplanes," in *Proc. IEEE Conference on Control Applications (CCA)*, Antibes/Nice, France, Oct. 2014, pp. 1003–1010.

[2] M. Brossard, J.-P. Condomines, and S. Bonnabel, "Tightly coupled navigation and wind estimation for mini UAVs," in *Proc. 2018 AIAA Guidance, Navigation, and Control Conference*, Kissimee, Florida, USA, Jan. 2018, p. 1843.

[3] J. N. Gross, Y. Gu, M. B. Rhudy, S. Gururajan, and M. R. Napolitano, "Flight-test evaluation of sensor fusion algorithms for attitude estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2128–2139, 2012.

[4] B. D. O. Anderson and J. B. Moore, "Optimal Filtering," *Dover Publications*, vol. 1, no. 1, p. 367, 1979.

[5] S. Leutenegger, "Unmanned solar airplanes - design and algorithms for efficient and robust autonomous operation," Ph.D. dissertation, ETH Zürich, Zürich, Switzerland, 2014.

[6] J. Solà, "Quaternion kinematics for the error-state Kalman filter," *CoRR*, vol. abs/1711.02508, 2017. [Online]. Available: http://arxiv.org/abs/1711.02508

[7] R. Brozel, "Total Energy Compensation in Practice," *ILEC GmbH*, 1985.

[8] P. Huang, H. Meyr, M. Dörpinghaus, and G. Fettweis, "Observability analysis of flight state estimation for UAVs and experimental validation," in *Proc. IEEE International Conference on Robotics and Automation (accepted)*, Paris, France, Jun. 2020.

[9] P. D. Groves, *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech house, 2013.

[10] D. A. Caughey, "Introduction to aircraft stability and control course notes for M&AE 5070," *Sibley School of Mechanical & Aerospace Engineering Cornell University*, 2011.

[11] D. W. Allan, "Statistics of atomic frequency standards," *Proceedings of the IEEE*, vol. 54, no. 2, pp. 221–230, 1966.

[12] A. Fallis, *Glider Flying Handbook*. U.S. Department of Transportation, 2013, vol. 53, no. 9.

[13] H. Reichman, "Cross-country soaring (Streckensegelflug)," 1978.

[14] G. Guennebaud, B. Jacob *et al.*, "Eigen v3," http://eigen.tuxfamily.org, 2010.