ICAS 2016
30th Congress of the International Council of the Aeronautical Sciences
DCC, Daejeon, Korea ; September 25-30, 2016

FLIGHT PATH RECONSTRUCTION FOR AN UNMANNED AERIAL VEHICLE USING LOW-COST SENSORS

C. Göttlicher* , **F. Holzapfel***
* **Institute of Flight System Dynamics (FSD), Technische Universität München (TUM), D-85748 Garching, Germany**

Keywords: *Flight Path Reconstruction, Rauch-Tung-Striebel Smoother, Low-Cost, Unmanned Aerial Vehicle*

Abstract

This paper shows how low-cost, off-the-shelf sensors can be used to reconstruct the position, velocity, and attitude of an aircraft. An extended Rauch-Tung-Striebel Smoother is used offline to extract state information from recorded raw data signals. Some of the inherently large errors of low-cost sensors are determined by means of a Maximum-Likelihood scheme.

The applicability of this approach is shown using simulated as well as real data from a low cost flying test-bed, equipped with sensors obtained from the RC model market. In both cases, results are very promising. The proposed algorithm thus constitutes a possibility of obtaining high accuracy estimates of the true state trajectories for analysis and modeling.

1 Introduction

Estimation of the exact position, velocity, and attitude of an aircraft is a task which is traditionally performed within an integrated navigation system in real-time [1]. This data can be used for feedback control or visualization to the pilot. Naturally, the obtained accuracy can be improved significantly if high-quality sensors are used.

However, with the rise of the Unmanned Aerial Vehicle (UAV) market, where usually only small, low-cost, and low-accuracy sensors are employed, new approaches have to be found. These sensors commonly have comparatively large random and bias errors. Nevertheless, interesting results in the field of system identification and control have been obtained [2, 3].

This paper illustrates a post-processing approach, which serves to obtain accurate estimates of the true position, velocity, and attitude, solely based on low-cost sensors. It combines the information of one complete flight together with a kinematic model of the aircraft to reconstruct the state at every sampled time instant. The obtained data can then be used for system identification, dynamic analysis, or control algorithm assessment.

In the context of system identification, this step is commonly called Flight Path Reconstruction (FPR) [4].

2 Hardware Setup

The test vehicle is the institute's low-cost fixed wing testbed, which is based on a RC model of the aerobatic aircraft Zivko Edge540 [3], with 1.21 meters span.

At the center of the onboard avionics is a STM32F4 processor on a discovery board, which features a 32-bit ARM Cortex-M4F core with internal floating point unit, running at 16 MHz. Programming was completely done in ANSI-C, with no operating system on the micro controller.

The set of external sensors consists of an InvenSense MPU-6000EVB 6-DOF Inertial Measurement Unit (IMU), a u-blox Neo-6M Global Navigation Satellite System (GNSS) Receiver and a Honeywell HMC5883L 3-axis magnetometer, together with an EXP Tech analog differential pressure sensor and pitot tube. The latter consti-

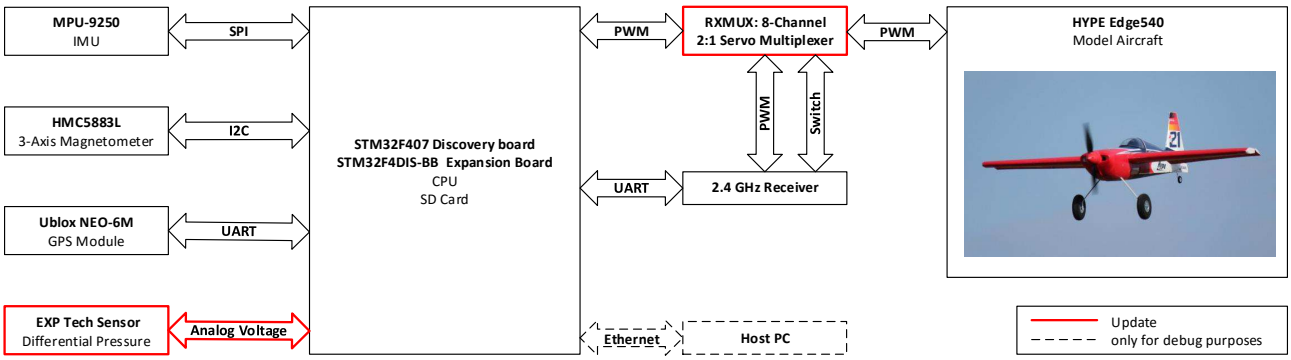

Fig. 1 Current setup of the low-cost flying testbed with modifications compared to [3]

tute updates compared to the setup described in [3]. Furthermore, the fail-safe capability of the setup is provided by the Acroname RxMux 2:1 8 Channel servo multiplexer, which can be used to bypass any automatic input generation in favor of direct pilot control.

The data, which is available for FPR is thus

- *inertial data:* rotational rates and accelerations at $100Hz$
- *GNSS data:* position and kinematic velocity at $5Hz$
- *magnetic data:* local magnetic field vector at $75Hz$
- *air data:* differential pressure at $100Hz$

All data is stored on a micro SD card for offline analysis. The hardware setup and testbed can be seen in figure 1, all electronics are powered via the central three cell lithium polymer battery.

3 Methods

Here, the general train of thoughts regarding FPR in [4] is followed with some modifications.

3.1 Extended Kalman Filter

For general, non-linear systems no globally valid filtering framework is available, only approximate solutions can be found. One of the most commonly used algorithms is the Extended Kalman Filter (EKF). It uses linearizations of the system equations around the currently best state estimate to be able to apply standard Kalman filter equations. A short summary of the algorithm is presented here, details can be found in [5, 6].

A mixed continuous-discrete formulation was chosen, with non-linear system dynamics of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\theta}\right) \tag{1}$$
$$\boldsymbol{y} = \boldsymbol{g}\left(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\theta}\right) \tag{2}$$

This formulation allows for non-linear inclusion of noise effects, as well as covering the influence of parameters $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$, which were tuned using a Maximum Likelihood (ML) like approach in order to improve the output fit, see section 3.6. The discrete system matrices were obtain using the partial derivatives

$$\boldsymbol{A}_k = \frac{\partial \boldsymbol{f}\left(\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{u}_k, \boldsymbol{0}, \boldsymbol{\theta}\right)}{\partial \boldsymbol{x}} \tag{3}$$
$$\boldsymbol{F}_k = \frac{\partial \boldsymbol{f}\left(\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{u}_k, \boldsymbol{0}, \boldsymbol{\theta}\right)}{\partial \boldsymbol{w}} \tag{4}$$
$$\boldsymbol{C}_k = \frac{\partial \boldsymbol{g}\left(\hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{u}_k, \boldsymbol{0}, \boldsymbol{\theta}\right)}{\partial \boldsymbol{x}} \tag{5}$$
$$\boldsymbol{G}_k = \frac{\partial \boldsymbol{g}\left(\hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{u}_k, \boldsymbol{0}, \boldsymbol{\theta}\right)}{\partial \boldsymbol{v}} \tag{6}$$

in order to compute the transition matrices

$$\boldsymbol{\Phi}_k = \exp\left(\boldsymbol{A}_k \cdot \Delta t\right) = \sum_{i=0}^{\infty} \frac{\left(\boldsymbol{A}_k \cdot \Delta t\right)^i}{i!}$$

$$\boldsymbol{\Gamma}_k = \int_0^{\Delta t} \exp\left(\boldsymbol{A}_k \tau\right) d\tau \cdot \boldsymbol{F}_k = \left(\sum_{i=1}^{\infty} \frac{\boldsymbol{A}_k^{i-1} \cdot \Delta t^i}{i!}\right) \cdot \boldsymbol{F}_k$$

Two noise sources are considered: measurement noise $\boldsymbol{v}_k \in \mathbb{R}^{n_v}$ and process noise $\boldsymbol{w}_k \in \mathbb{R}^{n_w}$, with the following characteristics

$$\begin{aligned} E\left[\boldsymbol{w}_k\right] &= \mathbf{0} & E\left[\boldsymbol{w}_k \boldsymbol{w}_j^\intercal\right] &= \delta_{kj}\boldsymbol{Q}_k \\ E\left[\boldsymbol{v}_k\right] &= \mathbf{0} & E\left[\boldsymbol{v}_k \boldsymbol{v}_j^\intercal\right] &= \delta_{kj}\boldsymbol{R}_k \\ & & E\left[\boldsymbol{w}_k \boldsymbol{v}_j^\intercal\right] &= \mathbf{0} \end{aligned}$$

3.1.1 *Propagation*

Propagation of the a priori state estimates $\hat{\boldsymbol{x}}_{k-1|k-1}$ was done non-linearly using a 4th order Runge-Kutta Scheme to solve

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k-1} &= \hat{\boldsymbol{x}}_{k-1|k-1} + \\ &\int_{t_k}^{t_{k+1}} \boldsymbol{f}\left(\hat{\boldsymbol{x}}_{k-1|k-1}, \tilde{\boldsymbol{u}}\left(\tau\right), \mathbf{0}, \boldsymbol{\theta}\right) d\tau \\ \tilde{\boldsymbol{u}}\left(\tau\right) &= \frac{t_{k+1}-\tau}{t_{k+1}-t_k}\boldsymbol{u}_k + \frac{\tau - t_k}{t_{k+1}-t_k}\boldsymbol{u}_{k+1} \end{aligned} \tag{7}$$

State covariance estimates were propagated using the linearized system dynamics

$$\begin{aligned} \boldsymbol{P}_{k|k-1}^{xx} &= \boldsymbol{\Phi}_{k-1}\boldsymbol{P}_{k-1|k-1}^{xx}\boldsymbol{\Phi}_{k-1}^\intercal \\ &+ \boldsymbol{\Gamma}_{k-1}\boldsymbol{Q}_{k-1}\boldsymbol{\Gamma}_{k-1}^\intercal \end{aligned} \tag{8}$$

Due to the relative simplicity of the model in section 3.5 a symbolic math engine was used to compute analytic derivatives for equations (3) - (6).

3.2 Correction

State and covariance correction were done linearly, while using the full non-linear output equation (2)

$$\begin{aligned} \boldsymbol{K}_k &= \boldsymbol{P}_{k|k-1}^{xx}\boldsymbol{C}_k^\intercal \\ &\cdot \left(\boldsymbol{C}_k\boldsymbol{P}_{k|k-1}^{xx}\boldsymbol{C}_k^\intercal + \boldsymbol{G}_k\boldsymbol{R}_k\boldsymbol{G}_k^\intercal\right)^{-1} \end{aligned} \tag{9}$$

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k|k-1} \\ &+ \boldsymbol{K}_k\left(\boldsymbol{z}_k - \boldsymbol{g}\left(\hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{u}_k, \mathbf{0}, \boldsymbol{\theta}\right)\right) \end{aligned} \tag{10}$$

$$\begin{aligned} \boldsymbol{P}_{k|k-1}^{xx} &= \left(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C}_k\right)\boldsymbol{P}_{k|k-1}^{xx}\left(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C}_k\right) \\ &+ \boldsymbol{K}_k\boldsymbol{G}_k\boldsymbol{R}_k\boldsymbol{G}_k^\intercal\boldsymbol{K}_k^\intercal \end{aligned} \tag{11}$$

Since magnetometer, GNSS and differential pressure measurements have different sample rates, not all elements of $\boldsymbol{z}_k$ are valid at all sampling instants k. Thus, at every instant t_k only the rows of $\boldsymbol{z}_k$, $\boldsymbol{g}$, $\boldsymbol{C}_k$, and $\boldsymbol{G}_k$, which actually contain valid measurements, are considered in equations (9) - (11).

3.3 Rauch Tung Striebel Smoother

To extract maximum information from the stored flight data, not only past but also future samples are used to estimate the current state. This is accomplished by optimally combining the results of a forward EKF pass, with a backwards EKF pass. One very efficient way to do this, is the Rauch-Tung-Striebel (RTS) smoother. Details can be found in [5, 7].

During the forward pass the following quantities have to be stored: $\hat{\boldsymbol{x}}_{k|k}$, $\hat{\boldsymbol{x}}_{k+1|k}$, $\boldsymbol{P}_{k|k}^{xx}$, $\boldsymbol{P}_{k+1|k}^{xx}$, and $\boldsymbol{\Phi}_k$, which are recursively combined to obtain the smoothed estimates ${}_s\boldsymbol{P}_k^{xx}$ and ${}_s\hat{\boldsymbol{x}}_k$

$$\begin{aligned} {}_s\boldsymbol{P}_N^{xx} &= \boldsymbol{P}_{N|N}^{xx} & k &= N-1\ldots 0 \\ {}_s\hat{\boldsymbol{x}}_N &= \hat{\boldsymbol{x}}_{N|N} \\ \boldsymbol{M}_k &= \boldsymbol{P}_{k|k}^{xx}\cdot\boldsymbol{\Phi}_k^\intercal\cdot\left(\boldsymbol{P}_{k+1|k}^{xx}\right)^{-1} \\ {}_s\boldsymbol{P}_k^{xx} &= \boldsymbol{P}_{k|k}^{xx} + \boldsymbol{M}_k\left({}_s\boldsymbol{P}_{k+1}^{xx} - \boldsymbol{P}_{k+1|k}^{xx}\right)\boldsymbol{M}_k^\intercal \\ {}_s\hat{\boldsymbol{x}}_k &= \hat{\boldsymbol{x}}_{k|k} + \boldsymbol{M}_k\left({}_s\hat{\boldsymbol{x}}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k}\right) \end{aligned}$$

By Incorporating the information of the complete timeseries, smoother estimates and lower state covariances can be accomplished compared to a pure forward pass.

Originally the RTS was developed for the linear Kalman filter case, however it can directly be applied to the EKF resulting in the Extended Rauch-Tung-Striebel (ERTS) smoother.

3.4 Sensor Errors

When working with low-cost components, it is paramount to explicitly consider sensor errors. Random errors are treated as zero-mean, white, Gaussian noise processes and taken care of by the EKF. In contrast to that, bias and scale factors are modeled as deterministic errors.

Inertial measurements are considered to be distorted by constant bias $\Delta\boldsymbol{a}$ and $\Delta\boldsymbol{\omega}$ and measurement noise $\boldsymbol{w}_{acc}$, $\boldsymbol{w}_{rot}$ and $\boldsymbol{w}_{rotacc}$

$$\left(\boldsymbol{a}_K^R\right)_{B,meas}^{II} = \left(\boldsymbol{a}_K^R\right)_B^{II} + \Delta\boldsymbol{a} + \boldsymbol{w}_{acc} \tag{12}$$

$$\left(\boldsymbol{\omega}_K^{IB}\right)_{B,meas} = \left(\boldsymbol{\omega}_K^{IB}\right)_B + \Delta\boldsymbol{\omega} + \boldsymbol{w}_{rot} \tag{13}$$

$$\left(\dot{\boldsymbol{\omega}}_K^{IB}\right)_{B,meas}^{B} = \left(\dot{\boldsymbol{\omega}}_K^{IB}\right)_B^{B} + \boldsymbol{w}_{rotacc} \tag{14}$$

Since rotational accelerations could not be measured directly, finite difference approximation

was used to obtain their value. Due to this bad approximation, the process noise covariance associated with them was assumed to be quite large.

GNSS position and velocity errors are are assumed to be purely random

$$\left(\boldsymbol{r}^R\right)_{WGS,meas} = \left(\boldsymbol{r}^R\right)_{WGS} + \boldsymbol{v}_{pos} \tag{15}$$

$$\left(\boldsymbol{v}_K^G\right)_{O,meas}^E = \left(\boldsymbol{v}_K^G\right)_O^E + \boldsymbol{v}_{vel} \tag{16}$$

Bias $\Delta\boldsymbol{b}$, a three component scale factor $\boldsymbol{K}_b$, and random errors $\boldsymbol{v}_{mag}$ are considered for the magnetometer

$$\begin{aligned}\left(\boldsymbol{b}\right)_{B,meas} = &\left(\boldsymbol{I}_3 + diag\left(\boldsymbol{K}_b\right)\right) \cdot \left(\boldsymbol{b}\right)_B \\ &+ \Delta\boldsymbol{b} + \boldsymbol{v}_{mag}\end{aligned} \tag{17}$$

The differential pressure errors are modeled as bias $\Delta\bar{q}$, scale factor $K_{\bar{q}}$, and measurement noise $v_{\bar{q}}$

$$\bar{q}_{meas} = \left(1 + K_{\bar{q}}\right)\bar{q} + \Delta\bar{q} + v_{\bar{q}} \tag{18}$$

The 2-norm of the attitude quaternions is introduced as additional pseudo measurement with value 1 in order to ensure proper quaternion scaling [8]

$$||\boldsymbol{q}_{OB}||_{meas} = ||\boldsymbol{q}_{OB}|| + v_q \tag{19}$$

3.5 Kinematic Equations

The 6 degree of freedom kinematic model is assembled from position, velocity and attitude propagation equations at the reference point R as illustrated in e.g. [1]. The system states are

$$\boldsymbol{x} = \left[\left(\boldsymbol{r}^R\right)_{WGS}^\intercal \ \left(\boldsymbol{v}_K^R\right)_B^{E,\intercal} \ \boldsymbol{q}_{BO}^\intercal \ \left(u_W^R\right)_O^E \ \left(v_W^R\right)_O^E\right]^\intercal$$

with position $\left(\boldsymbol{r}^R\right)_{WGS}$ in World Geodetic System 1984 (WGS84) coordinates, kinematic velocity $\left(\boldsymbol{v}_K^R\right)_B^E$ in body fixed coordinates, an attitude representation $\boldsymbol{q}_{BO}$ using unit-quaternions and horizontal wind components $\left(u_W^R\right)_O^E$ and $\left(v_W^R\right)_O^E$ in the North-East-Down (NED) frame.

The inputs are

$$\boldsymbol{u} = \left[\left(\boldsymbol{a}_K^R\right)_{B,meas}^{II,\intercal} \ \left(\boldsymbol{\omega}_K^{IB}\right)_{B,meas}^\intercal \ \left(\dot{\boldsymbol{\omega}}_K^{IB}\right)_{B,meas}^{B,\intercal}\right]^\intercal$$

with the measured accelerations, rotational rates and finite difference approximations for rotational accelerations in body fixed coordinates.

Finally, the outputs are

$$\begin{aligned}\boldsymbol{y} = \Big[&\left(\boldsymbol{r}^R\right)_{WGS,meas}^\intercal \ \left(\boldsymbol{v}_K^R\right)_{O,meas}^{E,\intercal} \\ &\left(\boldsymbol{b}\right)_{B,meas}^\intercal \ \bar{q}_{meas} \ ||\boldsymbol{q}_{OB}||_{meas}\Big]^\intercal\end{aligned}$$

with the position vector in WGS84 coordinates $\left(\boldsymbol{r}^R\right)_{WGS,meas}$, velocity in the NED frame $\left(\boldsymbol{v}_K^R\right)_{O,meas}^{E,\intercal}$, the measured magnetic field in the body fixed frame $\left(\boldsymbol{b}\right)_{B,meas}$, measured dynamic pressure $\bar{q}_{meas}$ and the 2-norm of the current attitude quaternion $||\boldsymbol{q}_{OB}||_{meas}$, as proposed in [8].

One part of the parameter vector for ML estimation comprises of the sensor errors

$$\boldsymbol{\theta}_{err} = \left[\begin{array}{cccccc}\Delta\boldsymbol{a}^\intercal & \Delta\boldsymbol{\omega}^\intercal & \Delta\boldsymbol{b}^\intercal & \boldsymbol{K}_b^\intercal & \Delta\bar{q} & K_{\bar{q}}\end{array}\right]^\intercal \tag{20}$$

3.5.1 State Equations

The non-linear state space model consists of kinematic relationships only, i.e. apart from sensor installation locations, no aircraft specific information is needed [9, 10].

A transformation matrix $\boldsymbol{M}_{BO}$ from local NED to body-fixed coordinates is assembled from the attitude quaternion $\boldsymbol{q}_{OB}$, see [1]. $\boldsymbol{M}_{BO}$ is then used to transform $\left(\boldsymbol{v}_K^R\right)_B^E$ from body fixed to NED coordinates

$$\left(\boldsymbol{v}_K^R\right)_O^E = \boldsymbol{M}_{BO}^\intercal \cdot \left(\boldsymbol{v}_K^R\right)_B^E \tag{21}$$

$\left(\boldsymbol{v}_K^R\right)_O^E$ is used for WGS84 position propagation

$$\left(\dot{\boldsymbol{r}}^R\right)_{WGS} = \begin{bmatrix} \frac{\left(u_K^R\right)_O^E}{M_\phi + h_{WGS}^R} \\ \frac{\left(v_K^R\right)_O^E}{N_\phi + h_{WGS}^R \cos\left(\phi_{WGS}^R\right)} \\ -\left(w_K^R\right)_O^E \end{bmatrix} + \boldsymbol{w}_{pos} \tag{22}$$

with WGS84 altitude h_{WGS}^R, latitude ϕ_{WGS}^R, and radii of curvature M_ϕ and N_ϕ. Even though the position propagation is a purely kinematic relationship, it was found that introducing a process noise term $\boldsymbol{w}_{pos}$ with small covariance improved filter results.

To obtain the velocity propagation equation, measured accelerations have to be corrected for Coriolis, rotational acceleration, centripetal and gravitational influences

$$\begin{aligned}\left(\dot{\boldsymbol{v}}_K^R\right)_B^{EB} = &\left(\boldsymbol{a}_K^R\right)_B^{II} - \left(\boldsymbol{\omega}_K^{IB}\right)_B \times \left(\boldsymbol{v}_K^R\right)_B^E \\ &- \left(\dot{\boldsymbol{\omega}}_K^{IB}\right)_B^B \times \left(\boldsymbol{r}^{R,IMU}\right)_B \\ &- \left(\boldsymbol{\omega}_K^{IB}\right)_B \times \left[\left(\boldsymbol{\omega}_K^{IB}\right)_B \times \left(\boldsymbol{r}^{R,IMU}\right)_B\right] \\ &+ \boldsymbol{M}_{BO}\boldsymbol{g}_O\end{aligned} \tag{23}$$

Inertial data is computed from measurements with the help of equations (12) - (14), which introduces process noise terms. $\left(\boldsymbol{r}^{R,IMU}\right)_B$ represents the installation location of the IMU with respect to the considered reference point R.

Attitude propagation is done using the quaternion propagation equation

$$\dot{\boldsymbol{q}}_{OB} = \frac{1}{2}\begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}\left(\boldsymbol{\omega}_K^{IB}\right)_B \tag{24}$$

where equation (13) is used to obtain $\left(\boldsymbol{\omega}_K^{IB}\right)_B$ from measurements.

A rough estimate of horizontal wind can be obtained by comparing kinematic velocities with measured differential pressure readings. To do so, horizontal wind components are modeled as random walk with

$$\begin{bmatrix} \left(\dot{u}_W^R\right)_O^{EO} \\ \left(\dot{v}_W^R\right)_O^{EO} \end{bmatrix} = \boldsymbol{w}_{wind} \tag{25}$$

Equations (22) - (25) are then assembled to obtain a system of the form (1).

3.5.2 Output Equations

For output estimates of measured position and velocity, equations (15), (16) and (21) can be used directly.

Magnetometer readings in body-fixed coordinates are modeled as a constant magnetic field vector $\left(\boldsymbol{b}\right)_O$ computed from the World Magnetic Model 2015 (WMM2015), and translated to body fixed coordinates

$$\left(\boldsymbol{b}\right)_B = \boldsymbol{M}_{BO}\left(\boldsymbol{b}\right)_O \tag{26}$$

Together with equation (17) output estimates can be obtained.

Dynamic pressure readings are computed as indicated airspeed from the velocity state, together with the horizontal wind estimates

$$\begin{aligned}\left(\boldsymbol{v}_A^R\right)_B^E &= \left(\boldsymbol{v}_K^R\right)_B^E - \boldsymbol{M}_{BO}\begin{bmatrix} \left(u_W^R\right)_O^E \\ \left(v_W^R\right)_O^E \\ 0 \end{bmatrix} \\ \bar{q} &= \frac{\rho_0}{2}|| \left(\boldsymbol{v}_A^R\right)_B^E ||^2\end{aligned} \tag{27}$$

with International Standard Atmosphere (ISA) air density at sea level of $1.225\frac{kg}{m^3}$. Differing density is lumped together with scale factor errors in $K_{\bar{q}}$. Output estimates can be assembled using equation (18).

Output estimates for the quaternion norm can directly be obtained from equation (19).

3.6 Parameter Estimation

With the test-setup at hand, flight duration is usually limited to about 6 minutes. During this time span, sensor errors are assumed to be constant and estimated offline using a ML like approach.

Classic ML methods can be rephrased as optimization problems that minimize the following cost-function [11, 4]

$$\begin{aligned}&\min_{\boldsymbol{\theta},\boldsymbol{\mathcal{B}}} J \\ J\left(\boldsymbol{\theta},\boldsymbol{\mathcal{B}}\right) &= \frac{1}{2}\sum_{k=0}^{N}\boldsymbol{r}_k^\intercal\boldsymbol{\mathcal{B}}^{-1}\boldsymbol{r}_k + \frac{N+1}{2}\ln|\boldsymbol{\mathcal{B}}| \\ \boldsymbol{r}_k &= \boldsymbol{z}_k - \boldsymbol{y}_k\left(\boldsymbol{\theta}\right)\end{aligned}$$

Due to the strong interconnection of the unknown parameter vector $\boldsymbol{\theta}$ and the unknown residual covariance matrix $\boldsymbol{\mathcal{B}}$ a common approach is to separate the optimization in two steps. First, the maximum likelihood estimate of $\boldsymbol{\mathcal{B}}$ is computed according to

$$\boldsymbol{\mathcal{B}} = \frac{1}{N+1}\sum_{k=0}^{N}\boldsymbol{r}_k\boldsymbol{r}_k^\intercal \tag{28}$$

which is then considered fixed during the computation of an update for $\boldsymbol{\theta}$. Those two steps are iterated until convergence [4].

Here, the problem arises, that not at every sampling instant valid measurements of all outputs are available which prohibits the computation of $\boldsymbol{\mathcal{B}}$ according to (28). A possible remedy is to assume zero residuals for invalid measurements

$$(\tilde{r}_i)_k = \begin{cases} (y_i)_k(\boldsymbol{\theta}) - (z_i)_k & \text{if } (z_i)_k \text{ is valid} \\ 0 & \text{otherwise} \end{cases} \tag{29}$$
$$i = 1 \dots n_y; \qquad k = 0 \dots N$$

An estimate of the residual covariance matrix can then be computed according to

$$\tilde{\boldsymbol{\mathcal{B}}} = \frac{1}{N+1} \sum_{k=0}^{N} \tilde{\boldsymbol{r}}_k \tilde{\boldsymbol{r}}_k^{\mathsf{T}} \tag{30}$$

The residual covariances are now underestimated, since less than $N+1$ valid samples are involved in their computation. Thus, $\tilde{\boldsymbol{\mathcal{B}}}$ is scaled before entering the cost function. The diagonal scaling matrix $\boldsymbol{W}$ is computed from the number of valid samples per output M_i

$$\boldsymbol{W} = diag\left(\sqrt{\frac{N}{M_1}} \dots \sqrt{\frac{N}{M_{n_y}}}\right) \tag{31}$$

The final cost function to be optimized is

$$\tilde{J}\left(\boldsymbol{\theta}, \tilde{\boldsymbol{\mathcal{B}}}\right) = \frac{1}{2} \sum_{k=0}^{N} \tilde{\boldsymbol{r}}_k^{\mathsf{T}} \left(\boldsymbol{W}\tilde{\boldsymbol{\mathcal{B}}}\boldsymbol{W}^{\mathsf{T}}\right)^{-1} \tilde{\boldsymbol{r}}_k + \frac{N+1}{2} \ln\left|\boldsymbol{W}\tilde{\boldsymbol{\mathcal{B}}}\boldsymbol{W}^{\mathsf{T}}\right| \tag{32}$$

which is minimized using the two-step procedure illustrated above.

Initial values for parameter estimates are computed in two steps. In a first step, while the aircraft is at rest, averaged accelerations are compared to their nominal value. The difference is attributed equally to all three bias elements in $\Delta\boldsymbol{a}$. Similarly, averaged rotational rates are compared to their nominal value of 0 to obtain an initial guess for $\Delta\boldsymbol{\omega}$.

In a second step, using the initial guesses for $\Delta\boldsymbol{a}$ and $\Delta\boldsymbol{\omega}$, the EKF is applied with only GNSS measurements for correction. Magnetometer and airspeed readings are computed afterwards from the estimated state trajectory to obtain estimates of $(\boldsymbol{b})_{B,k}$ and $\bar{q}_k$. Now, using measurements $(\boldsymbol{b})_{B,meas,k}$ and $\bar{q}_{meas,k}$, equations (17) and (18) can be reformulated as ordinary least squares problems for $k = 0 \dots N$ in the sensor error parameters $\Delta\boldsymbol{b}$ and $\boldsymbol{K}_b$, respectively $\Delta\bar{q}$, and $K_{\bar{q}}$. The solution is used as initial guess for the ML like estimation algorithm.

Most initial values $\boldsymbol{x}_0$ for the ERTSsmoother can be computed directly from the measurements, only initial values for attitude and wind states are included in the parameter vector $\boldsymbol{\theta}$ and thus estimated along with the sensor errors.

4 Results

Two different scenarios are investigated

- simulated data, for which true parameter, state and output values are known
- real data, which was gathered using the testbed described in section 2

4.1 Reconstruction Result

4.1.1 Simulated Data

The reference simulation data was generated by using the estimation result from the real data case as reference trajectory for a simulation model of a small scale, aerobatic aircraft. Simple cascaded PI-controllers were used to eventually obtain a trajectory, having roughly the same rate and attitude time histories as the measured data.

For most outputs, the same sensor error models as in 3.4 were used to create artificial measurements. Only the GNSS readings were not augmented with white, but filtered white noise, which results in a more realistic GNSS error. Since the simulation model operates in different speed regimes, measurement noise for the airspeed sensor was increased to result in a comparable signal to noise ratio.

One crucial point in Kalman Filtering is the choice of process and measurement noise covariance matrices. In this simulated example, tuning those matrices was achieved on the basis of the true covariance matrices of the sensor models.

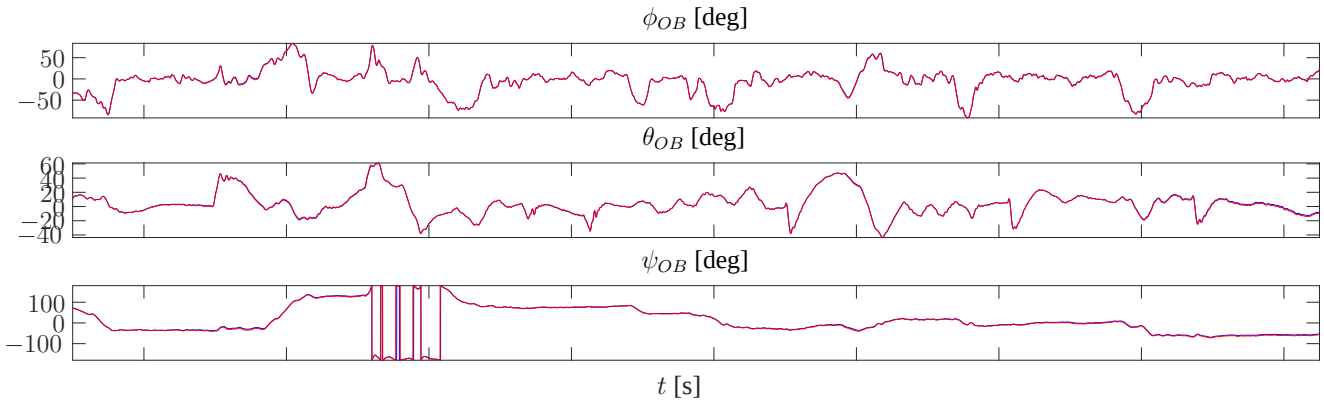

Fig. 2 Simulated (blue) and reconstructed attitude (red) represented as euler angles

The algorithms capability to reconstruct unmeasured states is illustrated in figure 2, where almost perfect attitude estimates are compared to the true time histories obtained during the simulation.

4.1.2 Real Data

Results from real measurement data can be seen in figure 5. Sensor covariances were tuned manually, based on some basic laboratory calibration. The output fit is somewhat worse compared to the simulation case, however the algorithm is still able to reconstruct a kinematically consistent trajectory, which fits the measurements well.

At $t = 172s$ there appears an error in the GNSS solution, as can be seen from the jump in position $(\phi)_{WGS84}$, $(\lambda)_{WGS84}$, and $(h)_{WGS84}$ in figure 5. However, the algorithm is able to reconstruct a smooth trajectory without jumps.

Due to the choice of covariance matrices, comparatively large discrepancies between measured and reconstructed values are tolerated in altitude $(h)_{WGS84}$ and vertical velocity $\left(w_K^R\right)_O^E$.

After correction of bias and scale factor errors, the magnetometer readings are very well reconstructed.

The resulting wind estimate can be seen in figure 3. Wind velocity in the range between $4\frac{m}{s}$ and $8\frac{m}{s}$ seems high. However, it was not a perfectly calm day, so the algorithm does not necessarily overestimate the true speed. Wind direction varies between 60 and 80 degrees, which seems reasonable, too.

4.2 Parameter Estimation Results

For the simulated data, a bar plot of the relative deviation $\left|\left(\hat{\theta}_i - \theta_i\right)/\theta_i\right|$ between estimated $\hat{\theta}_i$ and true parameter values θ_i can be seen in figure 4. Most parameter estimates lie very close to the respective true values. The relative deviation for sensor errors on the input side of the system is significantly larger than on the output side of the system. Especially accelerometer biases are not well captured, while magnetometer and differential pressure errors are estimated very well.

5 Discussion

5.1 Reconstruction

The algorithm performs very well in a simulation environment, as can be seen in figure 2 and 4. Thus, the presented approach of estimating constant sensor errors within an ERTS in order to reconstruct the true trajectory is, in principle, valid.

However, practical relevance of such an algorithm is only given if its applicability to real life data can be shown. Despite the low quality of the sensors, installed on the institute's testbed, good results were obtained as can be seen in the output fit in figure 5. Although the solution relies heavily on GNSS position and velocity, obvious errors like jumps can be overcome insofar as a smooth, kinematically consistent trajectory is reconstructed. Though the true flightpath is not known in this case, the reconstructed one seems realistic.

Due to the geometry of any GNSS solution, altitude is captured inherently worse compared to

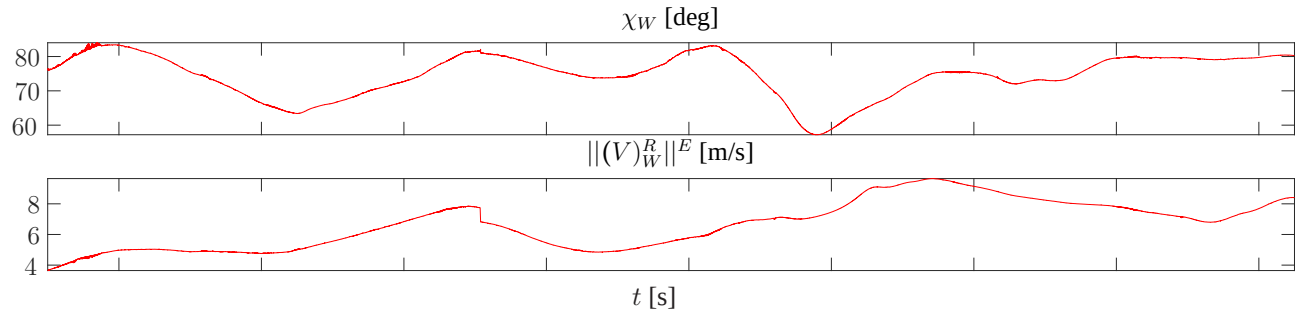

Fig. 3 Wind estimates for real data

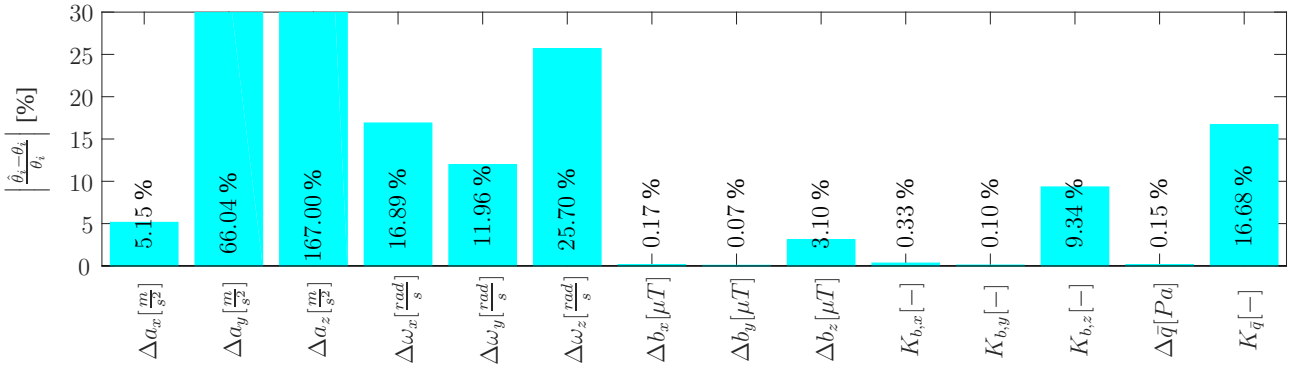

Fig. 4 Realtive parameter deviation

lateral and longitudinal position estimates. This is why larger errors in altitude $(h)_{WGS84}$ and vertical velocity $\left(w_K^R\right)_O^E$ are tolerated, compared to horizontal position and velocities. This error threshold is commonly set by the analyst via the corresponding covariances. A closer investigation of the relation between raw altitude and vertical speed measurements indicated that a low pass filter for $\left(w_K^R\right)_O^E$ seems to be implemented in the GNSS receiver. Thus, high frequency oscillations in $\left(w_K^R\right)_O^E$ are not captured very well, but can be reconstructed using the proposed algorithm. Since the true trajectory is unknown, no definite statement is possible, however the reconstructed output is at least consistent with inertial and magnetometer measurements.

In order to obtain a precise estimate of the aircraft's attitude, magnetometer readings turned out to be very useful. In contrast to IMU data, which is influenced by the trajectory's dynamics, the magnetometer offers a constant reference vector. Thus, after compensation for bias and scale factor, magnetometer data plays an important role in the reconstruction of the aircraft's attitude.

Due to the small scale and low speeds of the aircraft in use, already slight variations in aerodynamic velocity can have a significant influence on the trajectory, which is why an estimate of the wind conditions is very useful for further analysis. However, the wind estimate in figure 3 has to be treated with care. As no flow angle sensors where used, wind speed can only be estimated in the current body x-direction, which limits the overall accuracy. Furthermore, un-modeled effects like flow angle dependent errors of the pitot tube or lever arm effects due to its installation location will be lumped into the wind estimate. Nevertheless, the result can at least be used as a rough estimate for real conditions.

5.2 Parameter Estimation

Even though not all error parameters of the model can be estimated perfectly, as can be seen in figure 4, the overall performance of the algorithm is satisfactory. Especially because the main focus does not lie on the precise determination of sensor errors but on the reconstruction of true states and outputs.

In the case of real measurement data, several sensor errors remain un-modeled. However, the good output fit indicates that the class of mod-

els represented by the equations proposed here is versatile enough to cover the most important influencing factors, which eventually leads to a good output fit.

6 Conclusions

An approach to reconstruct estimates of the true state and output trajectory of an aircraft based on low-cost sensors has been illustrated. It is based on an ERTS smoother and ML like parameter estimation.

Several aspects show the validity and practical applicability of this approach. In a simulation environment, the algorithm is able to almost perfectly reconstruct true state trajectories and yields good parameter estimates. Furthermore, even with low-cost sensor data, the reconstructed outputs show good consistency compared to the measured quantities and a kinematically consistent trajectory can be reconstructed.

In the future, the incorporation of sensors for pressure altitude and static temperature could further improve the results. Also, more elaborate sensor error models can be implemented to e.g. cover lever arm effects of the Pitot system.

References

[1] Wendel J. *Integrierte Navigationssysteme: Sensordatenfusion, GPS und Inertiale Navigation*. Oldenbourg, R, München, 2., überarbeitete auflage edition, 2010.

[2] Dorobantu A, Johnson W, Lie F. A, Taylor B, Murch A, Yew Chai Paw , Gebre-Egziabher D, and Balas G. An airborne experimental test platform: From theory to flight. In *2013 American Control Conference (ACC)*, pages 659–673, 2013.

[3] Krause C, Göttlicher C, and Holzapfel F. Designing a low cost fixed wing flying testbed. In *Aerospace Electronics and Remote Sensing Technology (ICARES), 2014 IEEE International*. IEEE, 2015.

[4] Jategaonkar R. V. *Flight vehicle system identification: A time domain methodology*, volume v. 216 of *Progress in astronautics and aeronautics*. American Institute of Aeronautics and Astronautics, Reston, Va., 2006.

[5] Simon D. *Optimal state estimation: Kalman, H [infinity] and nonlinear approaches*. Wiley-Interscience, Hoboken, N.J., 2006.

[6] Einicke G. A. *Smoothing, filtering and prediction: Estimating the past, present and future*. In-Tech, Rijeka, 2012.

[7] Crassidis J. L and Junkins J. L. *Optimal estimation of dynamic systems*, volume 24 of *Chapman & Hall/CRC applied mathematics & nonlinear science*. CRC Press, Boca Raton, FL, 2nd ed. edition, 2012.

[8] Bar Itzhack I, Deutschmann J, and Markley F. Quaternion normalization in additive ekf for spacecraft attitude determination. In *Navigation and Control Conference*, 12 August 1991 - 14 August 1991.

[9] Stevens B. L and Lewis F. L. *Aircraft control and simulation*. J. Wiley, Hoboken, N.J., 2nd ed edition, 2003.

[10] Brockhaus R, Alles W, and Luckner R. *Flugregelung*. Springer Berlin, Berlin, 3., neu bearbeitete aufl. edition, 2010.

[11] Klein V and Morelli E. A. *Aircraft system identification: Theory and practice*. AIAA education series. American Institute of Aeronautics and Astronautics, Reston, VA, 2006.

Contact Author Email Address

The corresponding author's email address is `christoph.goettlicher@tum.de`

Copyright Statement

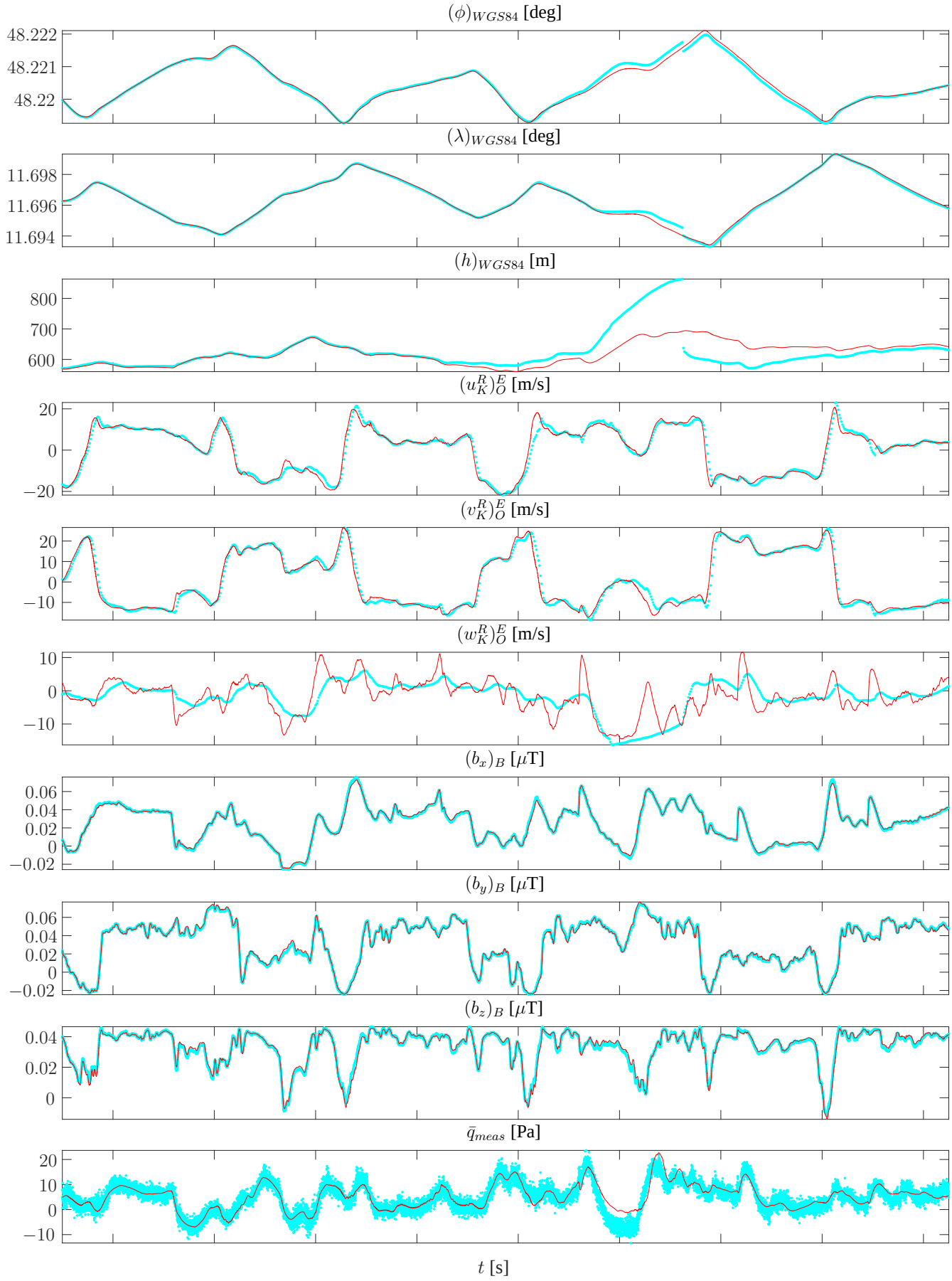

Fig. 5 Measured (cyan) and reconstructed (red) outputs for the real data case