A Nonlinear Filter for Efficient Attitude Estimation of Unmanned Aerial Vehicle (UAV)

Jarosław Gośliński¹ · Wojciech Giernacki² · Andrzej Królikowski²

Received: 10 January 2018 / Accepted: 17 October 2018 / Published online: 3 November 2018
© The Author(s) 2018

Abstract

Autonomous estimation of the state is of key importance in UAVs, as the measurement systems may experience faults and failures. Thus estimation techniques must provide estimates of the most important variables used in the control algorithms for safe, autonomous, unmanned flights. In this paper, a filter with low computational complexity for attitude estimation of a quadrotor UAV is introduced, with a model suitable for Fault-Tolerant Observation. The new filtration method, called the Square Root Unscented Complementary Kalman Filter (SRUCKF), is based on the commonly-known Kalman Filter (KF) in its nonlinear version, namely the Square Root Unscented Kalman Filter (SRUKF). The fundamental equation of the KF is modified so that the complementary feature of the filter is exalted. The new filter introduces characteristics that are analyzed on the basis of its application in quadrotor state estimation. Finally, the results are compared to an ordinary filter of the same type (using the Unscented Transformation). The presented studies indicate that the newly derived filter (SRUCKF) handles strong nonlinearities and gives results similar to those obtained from the SRUKF. Furthermore, it introduces lower computational burden, as the undergoing process uses diagonal matrices in its crucial places. In the paper, the estimation algorithms are tailored to a quadrotor UAV (Crazyflie 2.0), for which a quaternion-based model is proposed. The contribution of the paper lies in a Kalman-based novel state observer and its application in Fault-Tolerant Observation (FTO).

Keywords Nonlinear filter · Attitude estimation · Modified Kalman filter · Unmanned aerial vehicle · Unmanned quadrotor helicopter · Fault-tolerant control · Fault-tolerant observation

1 Introduction

Unmanned aerial vehicles have been an area of development. These flying vehicles have a wide range of applications and offer tremendous potential. Although UAVs are intentionally designed to fly without an operator, their autonomy is still strongly limited (among other reasons due to the use of imprecise low-cost sensors, which causes uncertainty in measurements of the current position and orientation). Therefore, rising UAVs to a higher level of autonomy is one of essential tasks of modern robotics. In order

to strengthen the robustness of the robot's measurements system, state observers are used.

In the past years, estimation in robotics was considered as a process of assessing unmeasurable variables of the studied object. At first, linear models were used in observers. This was possible thanks to the Luenberger observer [1], as well as the Kalman-Bucy filter [2]. Later on, a nonlinear version of the Kalman-Bucy filter was developed, namely the Extended Kalman Filter (EKF). The EKF works on a basis of the Kalman Filter theory, with the difference that in each step the nonlinear representation is linearized and then the Jacobian matrix is used as the system matrix. Although the EKF has been used for more than 40 years [3] and has been applied successfully in many works [4–6], it is an imprecise method. It suffers from approximations made during the linearization procedure. The EKF approximates the mean value and covariance up to the first order in the Taylor series. In the case of highly nonlinear problems it does not reflect the statistical values properly. It is also believed that in extreme cases it exhibits divergence and can be hard to tune when dealing with significant nonlinearities

✉ Wojciech Giernacki
wojciech.giernacki@put.poznan.pl

¹ AISENS, ul. F. Lubeckiego 23A, 60-348 Poznan, Poland

² Institute of Control, Robotics and Information Engineering, Electrical Department, Poznan University of Technology, Piotrowo 3A, 60-965 Poznan, Poland

[7]. The problem of filter divergence and linearization errors could be minimized in the next generation of filters which was initiated by the Second Order Extended Filter. The Second Order EKF uses linearization with accuracy up to the second term in the Taylor series. This allows precise estimation, however, the cost is high, as this filter uses the Jacobian and Hessians in the process. This complicates the implementation. Moreover, the second order EKF introduces higher computational complexity, which makes it less attractive. In the middle of the nineteen century, an interesting alternative for the EKF was developed: the Unscented Kalman Filter (UKF).

The Unscented Kalman Filter was firstly proposed in [8], as a solution to the EKF problems. The main concept behind the UKF is the Unscented Transformation (UT). During the Unscented Transformation, a set of carefully chosen points (vectors), called sigma points, is passed through the nonlinear function. The sigma points are chosen so that the mean value and covariance of the transformed function can be properly captured. The UT uses the resampling technique, so the statistic is calculated on the basis of the set of points. Technically, if the noise distribution is Gaussian, the UKF provides approximation up to the third order, and if the input is non-Gaussian, the filter still holds the approximation accuracy to at least the second order of the Tylor series [9]. The UKF has been applied in various works, focusing mainly on attitude estimation [10], position estimation [11], or model parameters estimation [12]. The algorithm is given in Appendix A.1. It uses the square root of the covariance matrix and the Cholesky decomposition. The square root is satisfied with a semi-positive matrix, however, the Cholesky factorization needs a positive-definite matrix to operate properly. This may lead to significant instability issues. Moreover, the basic square root of a given matrix $\mathbf{A}$ involves searching for $\mathbf{B}$:

$$\mathbf{A} = \mathbf{B}\mathbf{B}, \quad (1)$$

which is time consuming. Taking into consideration Lemma 1 and Lemma 2 (Appendix B), the following is obtained:

$$\mathbf{A} = \mathbf{Q}\mathbf{D}\mathbf{Q}^T, \quad (2)$$

$$\mathbf{B} = \mathbf{Q}\sqrt{\mathbf{D}}\mathbf{Q}^T, \quad (3)$$

where $\mathbf{Q}$ is orthogonal, while $\mathbf{D}$ is diagonal, which is proved in Appendix B. Matrices $\mathbf{Q}$ and $\mathbf{D}$ are found in the process of Schur decomposition [13]. This method can be accomplished by using QR decomposition in an iterative manner. At least a few (k) iterations are needed. Assuming that the QR is $\mathcal{O}(nl^2)$, for any diagonalizable matrix $\mathbf{M} \in \mathbb{R}^{n \times l}$ and $n \geq l$, and taking into account that $\mathbf{A} \in \mathbb{R}^{n \times n}$, the overall complexity will be $\mathcal{O}(kn^3)$ (technically $k \geq 8$). On the other hand, the Cholesky factorization is $\mathcal{O}(n^3/6)$. Finally, the UKF algorithm complexity is $\mathcal{O}(kn^3)$.

The complexity, as well as the stability issues due to round-off errors in the Cholesky factorization, prompted researchers to develop a better implementation of the UKF. An interesting solution was given in [9], namely the Square Root Unscented Kalman Filter (SRUKF). The idea behind the SRUKF is to create an efficient tool for unscented filtration, where the square-root of the matrix is replaced by a stable square root form propagated throughout the filtration process. It means that the square root is directly updated, without its straight calculation. To clarify this, the SRUKF algorithm is given in Appendix A.2. The algorithm uses the QR decomposition and the Cholesky update/downdate methods instead of square root and the Cholesky factorization. The Cholesky update is $\mathcal{O}(n^2)$ and the QR for that case (decomposed matrix $\in \mathbb{R}^{3n \times n}$) is $\mathcal{O}(3n^3)$. The advantage of the SRUKF over the UKF can be clearly seen. The authors of the SRUKF claim that the method is numerically stable and guarantees semi-positiveness of the square root of the covariance matrix [9]. The SRUKF did not, however, meet with enthusiastic reception. In fact, only a few works regarding the SRUKF were published in the first eight years after its disclosure. Later, the SRUKF was used in many applications, mainly in state estimation of mobile robots (for instance [14]).

Both algorithms (UKF and SRUKF) are suited for applications where the state and output functions are nonlinear, which is a general case. In many cases the output function $\mathbf{h}(\dots)$ is a linear combination of state elements. More often than not, the output vector uses the whole state vector. For these cases, the previously given methods can be tailored, yielding a better performance. Here, the authors propose a method which is based on the Kalman Filter theory, though it uses a modified fundamental equation. The method is suited for the particular case when the state and output vectors are identical, i.e.:

$$\mathbf{y}_k = \mathbf{x}_k. \quad (4)$$

In order to verify the proposed method, the problem of state estimation in a quadrotor UAV is studied. Although state estimation in UAVs might be considered as a separate research problem, it is highly important to show a nonlinear estimation problem and to propose tools handling the case. Among various estimation problems, the quadrotor state estimation is one of the most important research topics in the field of UAVs. In order to not get into details, only orientation will be studied in terms of estimation performance during normal and disturbed conditions. This often refers to fault-tolerant attitude estimation and fault-tolerant observation [15, 16]. Orientation estimation is the most common need which has to be fulfilled in all types of drones, as the provided data are necessary to fly (unlike position). Among many methods of attitude estimation, there are a few that are used in standalone devices called

the Attitude and Heading Reference System (AHRS), and those which are employed in on-board computers as algorithms. In the AHRS, the Strapdown Inertial Navigation Systems (SINS) are used. In a basic form, an SINS is a combination of two basic types of sensors: accelerometers and gyroscopes. By taking into account these two, angular velocity as well as angular position can be obtained. Although an SINS system made of a three-axis accelerometer and a three-axis gyroscope can provide data to calculate attitude and velocities, this is insufficient to calculate heading. In order to estimate full orientation, at least magnetometer data must be incorporated.

There are many algorithms and nonlinear techniques that can be used for attitude and position estimation on the basis of data from an SINS supported by magnetometers and GPS. The most popular are Kalman Filters (KF) [17], Complementary Filters (CF) [18], and Particle Filters (PF) [19]. They are, however, not necessarily proper algorithms in the case of attitude and position estimation of multirotors. As the environment of the on-board systems of UAV multirotors is rather hostile for sensitive MEMS sensors and permeated with distorted magnetic fields, special methods must be used for undisturbed estimation. Moreover, the estimation process must be robust and valid even in the case of sensors failure/fault.

In the paper [20] the authors present methods of state estimation. The most interesting is the AHRS-related part, where a new filter design is shown. The authors have focused on solving the Wahba's problem, which constitutes the merit of the orientation observer. The solution indeed proved to be reliable, even though when dynamic or magnetic disruptions occur. However, that is not due to the complementary filter but to the proposed model. Thus the complementary filter can be simply replaced with the Kalman Filter. Researchers care about low complexity and ease of tuning, especially when it comes to UAV's on-board implementations, but when one is searching for a minimal-variance solution, the Kalman Filter is a tool of choice. That is especially important when the estimation model is coupled with control, i.e. control has a direct influence on estimation and at the same time the estimation results are used in a control.

In this paper, authors propose modified SRUKF method called SRUCKF. The idea is to take into account a model of the quadrotor UAV and the AHRS data. Quadrotor UAV dynamics can be reflected as a nonlinear model with given parameters ($\mathbf{w}_k$). By providing a closed-loop model, where the input is the set position ($\mathbf{u}_k$), the model can be used to get the robot's attitude ($\mathbf{x}_k$). By making a fusion of the AHRS data and the model state, the robot state can be observed. This simple idea was tested in attitude estimation with the Unscented Kalman Filter in [21] and with the Adaptive Square-Root Unscented Kalman Filter (ASRUKF)

in [14]. In contrast, the new observer is compared to the standard SRUKF, and it is proved that the proposed method is similar in terms of overall estimation performance, with lower computational burden. The proposed method is shown to deal with serious AHRS errors due to a simulated fault/failure. The formulated Fault-Tolerant Observer can be used in sensor Fault-Tolerant Control tasks as well as in a robust state estimation framework. The provided method, i.e. the SRUCKF, the model, and the estimation architecture fall into the FTO and FTC discipline [22–24].

Results from the experiment are presented against the reference data gathered from the VICON motion capture system for the Crazyflie 2.0 unmanned aerial vehicle (Fig. 1). Details about Crazyflie 2.0 are widely presented in [25].

The most important, practical/useful advantage of the approach proposed here, in relation to the solutions available in commercial mid-range price quadrotors (like the popular PixHawk controller or XSens AHRS unit with stand-alone state estimation solutions already implemented in their firmware), is the openness and flexibility of the solution when the drone has to perform an unconventional task in very difficult conditions under variable load. Most of commercial units are still characterized by insufficient level of security due to insufficient quality and speed of the delivered position and orientation estimates, operating on various sensory sources (AHRS unit, GPS, laser and vision sensors) during autonomous flights between outdoor and indoor conditions. The UAV community is currently devoting more and more attention to research issues in the area of fault-tolerant observers [16] and fault-tolerant control [23, 24].

For example, let's consider mission no. 3 of the prestigious Mohamed Bin Zayed International Robotics Challenge (MBZIRC2020) in which we deal with the use of data fusion from various sensors to operate a

Fig. 1 Crazyflie 2.0 (a nano quadcopter equipped with 10-DOF IMU with accelerometer, gyro, magnetometer and pressure sensor, weight: 27 g, frame arm length: 6.48 cm)

UAV autonomously both indoors and outdoors, carrying a payload (fire extinguisher and blanket = flight with variable mass). The use of budget AHRS units here (usually supported by classical Kalman filtration) and low-level flight controllers (like the mentioned PixHawk) in combination with a basic GPS module is unfortunately insufficient to ensure fast and safe flights. The need to develop methods of precise estimation and control has already been demonstrated by the previous edition of the MBZIRC competition (mission with precise and fast autoland on a moving car [26] and "treasure hunt" [27], i.e. autonomous collection of metal discs by a group of UAVs). In the area of effective fault-tolerant estimation and fault-tolerant control (especially in visual estimation and control) there are many works published that draw inspiration from computational intelligence [15], as well as from the above-mentioned analytical estimation techniques.

The work presented here opens the door to the next level, in which we devote special attention to the problem of estimation in conditions of limited sensory information. Due to the multitude of issues and the potential volume of work, the authors decided to divide the results into a series of four articles dedicated to: UAV dynamics modeling [28], effective estimation methods, fault-tolerant estimation and control, as well as fusion of measurement and video data.

This article is organized as follows: in the second section the new method is derived, next, in the third section the estimation problem is explained, with a model of the quadrotor given in the quaternion form. In next two sections the tailored state observer is shown and state estimation results are presented. Finally, in the last section conclusions are drawn and a summary is given.

2 The New Observer

As has already been said, the new observer proposed in this paper is based on the Kalman Filter theory, however, its main equation is modified. We will start the new filter derivation process with a linear case. For the KF, the fundamental equation, given as follows:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}(\mathbf{y}_k - \hat{\mathbf{y}}_k), \quad (5)$$

is now replaced with the following, complementary one:

$$\hat{\mathbf{x}}_k = \mathbf{W}_k \hat{\mathbf{x}}_k^- + (\mathbf{I} - \mathbf{W}_k) \mathbf{y}_k. \quad (6)$$

It is highly important to note that the weight matrix $\mathbf{W}_k$ can be found in an optimization process, or in a similar way as it is done in the original filter. When the dimension of the output vector is equal to dimension of the state ($\dim(\mathbf{y}_k) = \dim(\mathbf{x}_k)$), Eqs. 7–8 represent a linear state-space model of the plant. It can actually be considered as an estimation

problem in itself, since the input $\mathbf{u}_k$ is acting on the plant and at the same time the vector $\mathbf{y}_k$ is being measured.

$$\mathbf{x}_k = \mathbf{A}_d \mathbf{x}_{k-1} + \mathbf{B}_d \mathbf{u}_{k-1} + \mathbf{v}_{k-1}, \quad (7)$$

$$\mathbf{y}_k = \mathbf{x}_k + \mathbf{o}_k, \quad (8)$$

where $\mathbf{v}_k \in \mathbb{R}^{n \times 1}$ is a zero mean white Gaussian process noise vector with covariance $\mathbf{Q}_k$, while $\mathbf{o}_k \in \mathbb{R}^{m \times 1}$ is a zero mean white Gaussian measurement noise vector with covariance $\mathbf{R}_k$. Since the model remains unchanged, the *a priori* covariance is the same as in the original KF:

$$\mathbf{P}_k^- = \mathbf{E} \left\{ \mathbf{e}_k^- (\mathbf{e}_k^-)^T \right\}, \quad (9)$$

$$\begin{aligned} \mathbf{e}_k^- &= \mathbf{x}_k - \hat{\mathbf{x}}_k^- \\ &= \mathbf{A}_d (\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}) + \mathbf{v}_{k-1}, \end{aligned} \quad (10)$$

$$\begin{aligned} \mathbf{P}_k^- &= \mathbf{E} \left\{ (\mathbf{A}_d \mathbf{e}_{k-1} + \mathbf{v}_{k-1}) (\mathbf{A}_d \mathbf{e}_{k-1} + \mathbf{v}_{k-1})^T \right\} \\ &= \mathbf{E} \left\{ (\mathbf{A}_d \mathbf{e}_{k-1} + \mathbf{v}_{k-1}) (\mathbf{e}_{k-1}^T \mathbf{A}_d^T + \mathbf{v}_{k-1}^T) \right\} \\ &= \mathbf{A}_d \mathbf{E} \left\{ \mathbf{e}_{k-1} \mathbf{e}_{k-1}^T \right\} \mathbf{A}_d^T + \mathbf{A}_d \mathbf{E} \left\{ \mathbf{e}_{k-1} \mathbf{v}_{k-1}^T \right\} \\ &\quad + \mathbf{E} \left\{ \mathbf{e}_{k-1} \mathbf{v}_{k-1}^T \right\} \mathbf{A}_d^T + \mathbf{E} \left\{ \mathbf{v}_{k-1} \mathbf{v}_{k-1}^T \right\} \end{aligned} \quad (11)$$

on the basis of:

$$\mathbf{P}_{k-1} = \mathbf{E} \left\{ \mathbf{e}_{k-1} \mathbf{e}_{k-1}^T \right\}. \quad (12)$$

$\mathbf{P}_{k-1}$ can be simplified to Eq. 13:

$$\mathbf{P}_k^- = \mathbf{A}_d \mathbf{P}_{k-1} \mathbf{A}_d^T + \mathbf{Q}_{k-1}. \quad (13)$$

The *a posteriori* error is derived on the basis of Eq. 6:

$$\begin{aligned} \mathbf{e}_k &= \mathbf{x}_k - \hat{\mathbf{x}}_k \\ &= \mathbf{x}_k - \mathbf{W}_k \hat{\mathbf{x}}_k^- - (\mathbf{I} - \mathbf{W}_k) \mathbf{y}_k \\ &= \mathbf{x}_k - \mathbf{W}_k \hat{\mathbf{x}}_k^- - (\mathbf{I} - \mathbf{W}_k) (\mathbf{x}_k + \mathbf{o}_k) \\ &= \mathbf{W}_k (\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - (\mathbf{I} - \mathbf{W}_k) \mathbf{o}_k \\ &= \mathbf{W}_k \mathbf{e}_k^- - (\mathbf{I} - \mathbf{W}_k) \mathbf{o}_k, \end{aligned} \quad (14)$$

while the *a posteriori* covariance is as follows:

$$\begin{aligned} \mathbf{P}_k &= \mathbf{E} \left\{ (\mathbf{x}_k - \hat{\mathbf{x}}_k) (\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \right\} \\ &= \mathbf{E} \left\{ [\mathbf{W}_k \mathbf{e}_k^- - (\mathbf{I} - \mathbf{W}_k) \mathbf{o}_k] [\mathbf{W}_k \mathbf{e}_k^- - (\mathbf{I} - \mathbf{W}_k) \mathbf{o}_k]^T \right\} \\ &= \mathbf{E} \left\{ \mathbf{W}_k \mathbf{e}_k^- (\mathbf{e}_k^-)^T \mathbf{W}_k^T - \mathbf{W}_k \mathbf{e}_k^- \mathbf{o}_k^T (\mathbf{I} - \mathbf{W}_k)^T + \right. \\ &\quad \left. - (\mathbf{I} - \mathbf{W}_k) \mathbf{o}_k (\mathbf{e}_k^-)^T \mathbf{W}_k^T + (\mathbf{I} - \mathbf{W}_k) \mathbf{o}_k \mathbf{o}_k^T (\mathbf{I} - \mathbf{W}_k)^T \right\} \quad (15) \\ &= \mathbf{W}_k \mathbf{E} \left\{ \mathbf{e}_k^- (\mathbf{e}_k^-)^T \right\} \mathbf{W}_k^T - \mathbf{W}_k \mathbf{E} \left\{ \mathbf{e}_k^- \mathbf{o}_k^T \right\} (\mathbf{I} - \mathbf{W}_k)^T + \\ &\quad - (\mathbf{I} - \mathbf{W}_k) \mathbf{E} \left\{ \mathbf{o}_k (\mathbf{e}_k^-)^T \right\} \mathbf{W}_k^T + \\ &\quad + (\mathbf{I} - \mathbf{W}_k) \mathbf{E} \left\{ \mathbf{o}_k \mathbf{o}_k^T \right\} (\mathbf{I} - \mathbf{W}_k)^T. \end{aligned}$$

Thus:

$$\begin{aligned} \mathbf{P}_k &= \mathbf{W}_k \mathbf{P}_k^- \mathbf{W}_k^T + (\mathbf{I} - \mathbf{W}_k) \mathbf{R}_k (\mathbf{I} - \mathbf{W}_k)^T \\ &= \mathbf{W}_k \mathbf{P}_k^- \mathbf{W}_k^T + \mathbf{R}_k - \mathbf{W}_k \mathbf{R}_k - \mathbf{R}_k \mathbf{W}_k^T + \mathbf{W}_k \mathbf{R}_k \mathbf{W}_k^T \quad (16) \\ &= \mathbf{R}_k - \mathbf{W}_k \mathbf{R}_k - (\mathbf{W}_k \mathbf{R}_k)^T + \mathbf{W}_k (\mathbf{P}_k^- + \mathbf{R}_k) \mathbf{W}_k^T. \end{aligned}$$

Note that $\mathbf{P}_k$ and $\mathbf{P}_k^-$ are symmetric, thus $\mathbf{P}_k = \mathbf{P}_k^T$ and $\mathbf{P}_k^- = (\mathbf{P}_k^-)^T$. By calculating $\mathbf{P}_k$ it is possible to determine the optimal filter gain $\mathbf{W}_k$. The procedure requires minimization of the trace of $\mathbf{P}_k$. By applying

Lemma 3 (Appendix B), the derivative of $\mathbf{P}_k$ can be calculated, yielding the following:

$$\frac{\partial \text{tr}(\mathbf{P}_k)}{\partial \mathbf{W}_k} = -2\mathbf{R}_k^T + 2\mathbf{W}_k(\mathbf{P}_k^- + \mathbf{R}_k), \quad (17)$$

which finally gives:

$$\mathbf{W}_k = \mathbf{R}_k(\mathbf{P}_k^- + \mathbf{R}_k)^{-1}. \quad (18)$$

With a given $\mathbf{W}_k$, $\mathbf{P}_k$ is rewritten to the Joseph form:

$$\begin{aligned} \mathbf{P}_k &= \mathbf{R}_k - \mathbf{W}_k\mathbf{R}_k - (\mathbf{W}_k\mathbf{R}_k)^T + \\ &\quad + \mathbf{R}_k(\mathbf{P}_k^- + \mathbf{R}_k)^{-1}(\mathbf{P}_k^- + \mathbf{R}_k)\mathbf{W}_k^T \\ &= \mathbf{R}_k - \mathbf{W}_k\mathbf{R}_k \\ &= (\mathbf{I} - \mathbf{W}_k)\mathbf{R}_k. \end{aligned} \quad (19)$$

Having already the covariance $\mathbf{P}_k$ and gain $\mathbf{W}_k$, the Complementary Kalman Filter equations are presented:

$$\hat{\mathbf{x}}_k^- = \mathbf{A}_k\hat{\mathbf{x}}_{k-1} + \mathbf{B}_k\mathbf{u}_{k-1}, \quad (20)$$

$$\mathbf{P}_k^- = \mathbf{A}_k\mathbf{P}_{k-1}\mathbf{A}_k^T + \mathbf{Q}_{k-1}, \quad (21)$$

$$\mathbf{W}_k = \mathbf{R}_k(\mathbf{P}_k^- + \mathbf{R}_k)^{-1}, \quad (22)$$

$$\hat{\mathbf{x}}_k = \mathbf{W}_k\hat{\mathbf{x}}_k^- + (\mathbf{I} - \mathbf{W}_k)\mathbf{y}_k, \quad (23)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{W}_k)\mathbf{R}_k. \quad (24)$$

This can be compared to the KF with $\mathbf{C}_k = \mathbf{I}_{n \times n}$:

$$\hat{\mathbf{x}}_k^- = \mathbf{A}_k\hat{\mathbf{x}}_{k-1} + \mathbf{B}_k\mathbf{u}_{k-1}, \quad (25)$$

$$\mathbf{P}_k^- = \mathbf{A}_k\mathbf{P}_{k-1}\mathbf{A}_k^T + \mathbf{Q}_{k-1}, \quad (26)$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{C}_k^T (\mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^T + \mathbf{R}_k)^{-1}, \quad (27)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{x}}_k^-), \quad (28)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k) \mathbf{P}_k^-. \quad (29)$$

These two basic filters, i.e. the proposed one (20)–(24) and the classic one (25)–(29), are used in the further derivation, as they constitute an analogy. The original covariances of the UKF, i.e. $\mathbf{P}_{xy}$ and $\mathbf{P}_{yy}$, can be denoted as follows: $\mathbf{P}_{xy} = \mathbf{P}_k^- \mathbf{C}_k^T$, $\mathbf{P}_{yy} = \mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^T + \mathbf{R}_k$, which explains the Eqs. 114 and 116.

Now, on that basis the equations of the UKF with linear output can be stated. The first equations from prediction (105)–(108), as well as the fundamental Eq. 115, do not change. The sigma points in the correction phase are not needed, so Eqs. 109–111 are no longer used, while Eqs. 112–114 and 116 are modified in the following way:

$$\mathbf{P}_{xy} = \mathbf{P}_k^-, \quad (30)$$

$$\mathbf{P}_{yy} = \mathbf{P}_k^- + \mathbf{R}_k, \quad (31)$$

$$\mathbf{K}_k = \mathbf{P}_{xy} \mathbf{P}_{yy}^{-1}, \quad (32)$$

$$\begin{aligned} \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_{yy} \mathbf{K}_k^T \\ &= (\mathbf{I} - \mathbf{K}_k) \mathbf{P}_{xy}. \end{aligned} \quad (33)$$

The new observer has two major differences (based on the filters analogy), i.e.:

$$\mathbf{P}_{xy} = \mathbf{R}_k \quad (34)$$

and

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{W}_k) \mathbf{R}_k, \quad (35)$$

which finally results in the following equations of the Unscented Complementary Kalman Filter (UCKF):

$$\chi_{k-1}^{a,i} = \hat{\mathbf{x}}_{k-1} + \text{sign}(n-i) \left(\sqrt{(n+\lambda)\mathbf{P}_{k-1}} \right)_i^T, \quad (36)$$

$$\chi_{k-1}^i = \mathbf{f}(\chi_{k-1}^{a,i}, \mathbf{u}_{k-1}), \quad (37)$$

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2n} W_i^{(m)} \chi_{k-1}^i, \quad (38)$$

$$\mathbf{P}_k^- = \sum_{i=0}^{2n} W_i^{(c)} (\chi_{k-1}^i - \hat{\mathbf{x}}_k^-) (\chi_{k-1}^i - \hat{\mathbf{x}}_k^-)^T + \mathbf{Q}_{k-1}, \quad (39)$$

$$\mathbf{P}_{xy} = \mathbf{R}_k, \quad (40)$$

$$\mathbf{P}_{yy} = \mathbf{P}_k^- + \mathbf{R}_k, \quad (41)$$

$$\mathbf{W}_k = \mathbf{P}_{xy} \mathbf{P}_{yy}^{-1}, \quad (42)$$

$$\hat{\mathbf{x}}_k = \mathbf{W}_k \hat{\mathbf{x}}_k^- + (\mathbf{I} - \mathbf{W}_k) \mathbf{y}_k, \quad (43)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{W}_k) \mathbf{P}_{xy}. \quad (44)$$

Applying the same rule to the SRUKF, the SRUCKF equations are obtained:

$$\chi_{k-1}^{a,i} = \hat{\mathbf{x}}_{k-1} + \text{sign}(n-i) \sqrt{(n+\lambda)} (\mathbf{S}_{k-1})_i^T, \quad (45)$$

$$\chi_{k-1}^i = \mathbf{f}(\chi_{k-1}^{a,i}, \mathbf{u}_{k-1}) \quad (46)$$

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2n} W_i^{(m)} \chi_{k-1}^i, \quad (47)$$

$$\mathbf{S}_k^- = \text{qr} \left\{ \begin{bmatrix} \sqrt{W_i^{(c)}} \chi_{k-1}^i - \hat{\mathbf{x}}_k^- & \sqrt{\mathbf{Q}_{k-1}} \end{bmatrix}^T \right\} \quad (48)$$

where $i \in (< 0 : 2n > \setminus n)$,

$$\mathbf{S}_k^- = \text{cholupdate} \left\{ \mathbf{S}_k^-, \chi_{k-1}^n - \hat{\mathbf{x}}_k^-, W_n^{(c)} \right\} \quad (49)$$

$$\mathbf{S}_y = \text{cholupdate} \left\{ \mathbf{S}_k^-, \left(\sqrt{\mathbf{R}_k} \right)_i, 1 \right\}, \quad (50)$$

$$\mathbf{P}_{xy} = \mathbf{R}_k, \quad (51)$$

$$\mathbf{W}_k = (\mathbf{P}_{xy} / \mathbf{S}_y^T) / \mathbf{S}_y, \quad (52)$$

$$\mathbf{x}_k = \mathbf{W}_k \mathbf{x}_k^- + (\mathbf{I} - \mathbf{W}_k) \mathbf{y}_k, \quad (53)$$

$$\mathbf{U} = \mathbf{W}_k \mathbf{R}_k, \quad (54)$$

$$\mathbf{S}_k = \text{cholupdate} \{ \mathbf{R}_k, \mathbf{U}_i, -1 \}, \quad (55)$$

where $i \in (< 1 : n >)$.

The new observers have the same effectiveness as the UKF and SRUKF, however, with better computational

properties.¹ The major advantage of the Complementary Kalman Filter lies in the relation $\mathbf{P}_{xy} = \mathbf{R}_k$, which guarantees a diagonal form of $\mathbf{P}_{xy}$. The SRUKF and SRUCKF will be studied in terms of errors and robustness. As it was stated before, neither the UKF nor the UCKF will be used, as these observers have worse properties compared to their Square-Root versions.

3 UAV Model

In the paper, the new observer SRUCKF and the standard SRUKF are studied in quadrotor UAV fault-tolerant state estimation. The problem of state estimation is one of the most important, especially when the on-board measurement system introduces serious errors. Moreover, the UAV's navigational modules may undergo faults and failures, which is the a issue in autonomous operation. The idea is to minimize the influence of these errors, faults, or even failures, while preserving functionality and performance of the control algorithms. This can be done via estimation, where data from more than one source of information are used. In one of the available solutions, at least two sensors of the same type are used. In another solution, a mathematical model as well as one sensor can be used. Here, the second idea is applied, along with a mathematical model of a quadrotor UAV. The basic model has two main parts: the orientation and the position, which together form the pose of the robot or the full state. Here, the model (adapted from [28]) is based on the Newton-Euler equations for rigid body in 3D space and has the following layout:

$$\dot{\omega} = \mathbf{I}^{-1} (\tau - \omega \times (\mathbf{I}\omega) - \tau_g - \tau_d - \tau_t), \quad (56)$$

$$\dot{\mathbf{v}} = \frac{1}{m} (\mathbf{f} + \mathbf{f}_g - \mathbf{f}_c - \mathbf{f}_d - \mathbf{f}_t), \quad (57)$$

where the torques are as follows:

$$\tau = \sum_{i=1}^n \mathbf{p}_i \times \mathbf{f}_i, \quad (58)$$

$$\tau_g = \omega \times (\mathbf{I}_w \omega_w), \quad (59)$$

$$\tau_d = \sum_{i=1}^n \mathbf{p}_i \times \mathbf{f}_{d_i}, \quad (60)$$

$$\tau_t = h_g \omega + b_w \text{diag}(\text{sign}(\omega)) \text{diag}(\omega) \omega + c_w \text{diag}(\omega)^2 \omega, \quad (61)$$

while the forces:

$$\mathbf{f} = \sum_{i=1}^n \mathbf{f}_i, \quad (62)$$

$$\mathbf{f}_g = m \mathbf{R}_e^b \mathbf{g}, \quad (63)$$

$$\mathbf{f}_c = -2m\omega \times \mathbf{v}, \quad (64)$$

$$\mathbf{f}_d = \sum_{i=1}^n \mathbf{f}_{d_i}, \quad (65)$$

$$\mathbf{f}_t = b_{w1} \text{diag}(\mathbf{v}) \mathbf{v}, \quad (66)$$

$$\mathbf{f}_{d_i} = \|\mathbf{f}_i\| (\mathbf{d}_{\Delta_i} + \mathbf{d}_{I_i} + \mathbf{d}_{T_i} + \mathbf{d}_{P_i}), \quad (67)$$

$$\mathbf{f}_i = [0 \ 0 \ b_i \omega_i^2]^T. \quad (68)$$

Nomenclature: ω – angular velocity, $\mathbf{I}$ – inertia matrix, τ – input torque (from propellers), τ_g – gyroscope torque due to the robot's frame and propulsion system, τ_d – torque from aerodynamic effects, τ_t – friction torque, $\mathbf{v}$ – linear velocity, m – UAV mass, $\mathbf{f}$ – input force, $\mathbf{f}_g$ – gravitational force, $\mathbf{f}_c$ – Coriolis force, $\mathbf{f}_d$ – drag force, $\mathbf{f}_t$ – friction force, $\mathbf{p}_i$ – position of the i -th propeller, $\mathbf{I}_w$ – inertia tensor of motor and propeller, ω_w – sum of propeller velocities, $\mathbf{R}$ – rotation matrix, h_g, b_w, c_w, b_{w1} – respectively: friction coefficients.

Note that the gravitational acceleration $\mathbf{g}$ depends on the reference system in which the positioning is done. The aerodynamic drags, i.e. flapping, induced, translational, and profile, acting on the i -th rotor, are denoted as $\mathbf{d}_{\Delta_i}, \mathbf{d}_{I_i}, \mathbf{d}_{T_i}, \mathbf{d}_{P_i}$, respectively. For the sake of clarity, further details will not be given, although the reader can find more information on aerodynamic drags in [29]. The studied quadrotor has a classic frame and is equipped with four rotors, its scheme is given in Fig. 2.

As it can be seen, two coordinate systems are introduced. The first one, namely the Body Frame $\{\mathcal{B}\mathcal{F}\}$, is fixed to the robot, while the second one is the inertial reference system $\{\mathcal{E}\mathcal{F}\}$. $\{\mathcal{E}\mathcal{F}\}$ is ENU, where the X axis is pointing East, Y is aligned with the North, and Z is pointing Up of the Earth geographic coordinates. This reference is popular, as it provides an intuitive insight. Since $\mathbf{g} \in \{\mathcal{E}\mathcal{F}\}$, it is given as follows:

$$\mathbf{g} = [0 \ 0 \ -g]^T. \quad (69)$$

The transformation matrix $\mathbf{R}_e^b$ transfers any given vector from $\{\mathcal{E}\mathcal{F}\}$ to $\{\mathcal{B}\mathcal{F}\}$, while its transpose $\mathbf{R}_b^e$ ensures the inverse action, i.e. $\{\mathcal{B}\mathcal{F}\} \rightarrow \{\mathcal{E}\mathcal{F}\}$. $\mathbf{R}_e^b$ is based on the Euler/Tait-Bryan angles or quaternions, although different representations can be also used. In the paper, quaternions are used, as they do not suffer from singularities or Gimbal-Lock issues.

If the arbitrary vector $\mathbf{v}^b \in \{\mathcal{B}\mathcal{F}\}$ is transferred to $\{\mathcal{E}\mathcal{F}\}$, $\mathbf{R}_b^e$ is used:

$$\mathbf{v}^e = \mathbf{R}_b^e \mathbf{v}^b. \quad (70)$$

¹Note that the CKF is equal to KF when $\mathbf{W}_k = \mathbf{I} - \mathbf{K}_k$, so the same UCKF and SRUCKF

Fig. 2 Quadrotor scheme

The same functionality is achieved with the quaternion $\mathbf{q}$ and its conjugate $\mathbf{q}^*$:

$$\tilde{\mathbf{v}}^e = \mathbf{q} \otimes \tilde{\mathbf{v}}^b \otimes \mathbf{q}^*, \quad (71)$$

or conversely:

$$\mathbf{v}^b = (\mathbf{R}_b^e)^T \mathbf{v}^e = \mathbf{R}_e^b \mathbf{v}^e, \quad (72)$$

$$\tilde{\mathbf{v}}^b = \mathbf{q}^* \otimes \tilde{\mathbf{v}}^e \otimes \mathbf{q}. \quad (73)$$

Note that $\tilde{\mathbf{v}} \in \mathbb{R}^4$, and in the case of quaternions for any vector $\mathbf{v}$ its extended version is equal to $\tilde{\mathbf{v}} = [0 \ \mathbf{v}]^T$.

The notation with Kronecker products ($\otimes$) is inconvenient, especially when performing multiple operations on a single vector. In order to make the quaternion notation clear and obvious to the reader, the authors decided to use the notation given in [30]. A basic explanation on the notation is given in Lemma 4 (Appendix B). By taking this into account, it is possible to express (71) as follows:

$$\tilde{\mathbf{v}}^e = \mathbf{q}^{-T} \mathbf{q}^+ \tilde{\mathbf{v}}^b, \quad (74)$$

or Eq. 73 as:

$$\tilde{\mathbf{v}}^b = \mathbf{q}^+ \mathbf{q}^- \tilde{\mathbf{v}}^e. \quad (75)$$

The orientation model can be constructed on the basis of the angular velocity equation given in quaternion representation [31]:

$$\tilde{\omega} = 2\mathbf{q}^* \otimes \dot{\mathbf{q}}, \quad (76)$$

where:

$$\tilde{\omega} = [0 \ \omega]^T, \quad (77)$$

which can be also given in the matrix form:

$$\tilde{\omega} = -2\mathbf{q}^- \dot{\mathbf{q}}^* = -2\mathbf{q}^+ \dot{\mathbf{q}} = 2\mathbf{q}^+ \dot{\mathbf{q}} = 2\mathbf{q}^- \dot{\mathbf{q}}^*. \quad (78)$$

The angular acceleration in $\{\mathcal{BF}\}$ is a derivative of $\tilde{\omega}$:

$$\ddot{\omega} = 2(\mathbf{q}^* \otimes \ddot{\mathbf{q}} + \dot{\mathbf{q}}^* \otimes \dot{\mathbf{q}}), \quad (79)$$

which, substituted to Eq. 56, yields:

$$\ddot{\tau} = 2\tilde{\mathbf{I}} \left(\mathbf{q}^+ \ddot{\mathbf{q}} + \mathbf{q}^+ \dot{\mathbf{q}} \right) + \mathbf{q}^+ \tilde{\omega} \tilde{\omega} + \mathbf{q}^+ \tilde{\omega} \tilde{\omega}_w + \ddot{\tau}_d + \ddot{\tau}_t, \quad (80)$$

$$\ddot{\mathbf{q}} = \frac{1}{2} \mathbf{q}^+ \mathbf{I}^{-1} \left(\ddot{\tau} - \mathbf{q}^+ \tilde{\omega} \tilde{\omega} - \mathbf{q}^+ \tilde{\omega} \tilde{\omega}_w - \ddot{\tau}_d - \ddot{\tau}_t \right) - \mathbf{q}^+ \mathbf{q}^+ \dot{\mathbf{q}}. \quad (81)$$

The same can be applied to the force equation, where the linear acceleration given in $\{\mathcal{EF}\}$ is as follows:

$$\ddot{\mathbf{v}}^e = \frac{1}{m} \left(\mathbf{q}^+ \mathbf{q}^+ (\ddot{\mathbf{f}} - \ddot{\mathbf{f}}_{co} - \ddot{\mathbf{f}}_d - \ddot{\mathbf{f}}_t) - \ddot{\mathbf{g}} \right). \quad (82)$$

Note that in Eqs. 80, 81 the inertia matrices have the form proposed in [31]:

$$\tilde{\mathbf{I}} = \begin{bmatrix} \frac{1}{2} \text{tr}(\mathbf{I}) & \mathbf{0}^T \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \tilde{\mathbf{I}}_w = \begin{bmatrix} \frac{1}{2} \text{tr}(\mathbf{I}_w) & \mathbf{0}^T \\ \mathbf{0} & \mathbf{I}_w \end{bmatrix}. \quad (83)$$

Since all vectors are extended with additional dimension, cross products are replaced by Kronecker products.

4 Tailored State Observer

The state observer is model-based. In the first phase it uses model data, and during the update phase it incorporates measurements. Here, a quaternion-based model is used, together with an AHRS.

Either a closed-loop model or an open-loop model may be used, although in the paper the research is narrowed only to the closed-loop model, as only the set point of position is available. In the open-loop model some physical quantities, e.g. rotor velocities, are needed, and in the case of the closed-loop model the set value of the real robot must be provided. Estimation with the closed-loop model can be realized as shown in Fig. 3. The scheme is proposed by the authors and will be studied in the next section. The most important idea behind the estimation scheme given in Fig. 3 lies in two control loops: one on the real robot and one using the robot model. The input is the same for these two loops and it is the set position. The control loop (either the robot

Fig. 3 Closed-loop estimation scheme

loop or the model loop) is actually composed of two loops: inner one and outer one, as depicted in Fig. 4. Note that quaternions are used directly in the attitude control.

The attitude control utilized a simple formula for the error, i.e.:

$$\mathbf{q}_{err} = \mathbf{q}_m^* \otimes \mathbf{q}_{set} \quad (84)$$

$$= \mathbf{q}_m^T \mathbf{q}_{set}, \quad (85)$$

which was proposed by [32]. $\mathbf{q}_{set}$ is calculated on the basis of set position. When controlling in $\{\mathcal{EF}\}$, the set quaternion is derived on the basis of the set Euler angles Θ_{set} :

$$\begin{aligned} \Theta_{set} &= \mathbf{R}_s \mathbf{u}_s \\ &= [\phi_{set} \theta_{set} \psi_{set}]^T, \end{aligned} \quad (86)$$

$$\mathbf{u}_s = [u_x u_y u_z]^T, \quad (87)$$

where $\mathbf{R}_s$ is determined on the basis of the chosen location of $\{\mathcal{BF}\}$, here as follows:

$$\mathbf{R}_s = \begin{bmatrix} \sin(\psi_s) & -\cos(\psi_s) & 0 \\ \cos(\psi_s) & \sin(\psi_s) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (88)$$

Note that $\psi_s = \psi$. The vector $\mathbf{u}_s$ is composed of axial values used directly by the position controllers.

Finally, the signal mixer, which is used in the motor controllers, has to be described. Normally, proper rotor velocities are increased and decreased, however, due to the

Fig. 4 Control scheme with inner and outer loops

Z axis position controller, an additional value ($t_k = thrust$) is equally added to all rotor velocities, forming the relation given in Eq. 89.

$$\mathbf{w}_k = \mathbf{t}_k + \mathbf{P} \mathbf{u}_k, \quad (89)$$

$$\mathbf{u}_k = [a_s \ b_s \ c_s]^T, \quad (90)$$

where $\mathbf{P}$ is a permutation matrix which depends on the location of rotors in $\{\mathcal{BF}\}$. Similar to $\mathbf{u}_s$, $\mathbf{u}_k$ are the axial set values provided by the attitude controllers.

Having covered the control tasks of the model (ie. from trajectory planning in the high-level control layer, through position and attitude control, to rotational speed control of UAV's motors in the low-level layer), observers can be derived. The control is performed according to Fig. 4. As it has already been stated, the observers use a model with controllers, forming a closed-loop system (see Fig. 3). The input for the model is the set position, while the output may consist of the full state, as well as of its selected elements. When dealing with a multirotor UAV, the most wanted are the orientation and position. When only the AHRS system is available, the position cannot be measured, though it can be simulated in the control process (via estimation using a model). Having such estimation data and the model, one is able to simulate a drone flight. On the other hand, the orientation is provided by the AHRS and it can be corrupted during faults and failures. That is why state estimators are used. Here, the authors decided to observe the quaternion, its first derivative and angular/linear velocity. For this purpose the state and output vectors are set as follows:

$$\hat{\mathbf{x}}_k = [\hat{\mathbf{q}}_k^T \ (\hat{\mathbf{v}}_k^b)^T \ \hat{\mathbf{q}}_k^T]^T, \quad (91)$$

$$\hat{\mathbf{y}}_k = [\hat{\mathbf{q}}_k^T \ (\hat{\omega}_k^b)^T]^T. \quad (92)$$

The functions $\mathbf{f}(\hat{\mathbf{x}}_k, \mathbf{u}_k)$ and $\mathbf{h}(\hat{\mathbf{x}}_k)$ have the following forms:

$$\begin{aligned} \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}) &= \\ & \begin{bmatrix} \hat{\mathbf{q}}_{k-1} + \frac{T}{2} \hat{\mathbf{q}}_{k-1}^+ \mathbf{I}^{-1} \mathbf{s}_{k-1} - T \hat{\mathbf{q}}_{k-1}^+ \left(\hat{\mathbf{q}}_{k-1}^+ \right)^T \hat{\mathbf{q}}_{k-1} \\ \check{\mathbf{v}}_{k-1}^b + \frac{T}{m} t_{k-1} \\ \hat{\mathbf{q}}_{k-1} + T \hat{\mathbf{q}}_{k-1}^+ \end{bmatrix}, \end{aligned} \quad (93)$$

where:

$$\begin{aligned} \mathbf{s}_{k-1} &= \check{\mathbf{t}}_{k-1} - \overset{+}{\omega}_{k-1} \check{\mathbf{I}} \check{\omega}_{k-1} - \overset{+}{\omega}_{k-1} \check{\mathbf{I}}_w \check{\omega}_{w_{k-1}} + \\ & - \check{\mathbf{t}}_{d_{k-1}} - \check{\mathbf{t}}_{t_{k-1}}, \end{aligned} \quad (94)$$

$$\begin{aligned} t_{k-1} &= \check{\mathbf{f}}_{k-1} - \overset{+}{\hat{\mathbf{q}}}_{k-1} \overset{-}{\hat{\mathbf{q}}}_{k-1} \check{\mathbf{g}} + 2m \overset{+}{\omega}_{k-1} \check{\mathbf{v}}_{k-1} + \\ & - \check{\mathbf{f}}_{d_{k-1}} - \check{\mathbf{f}}_{t_{k-1}}, \end{aligned} \quad (95)$$

and

$$\mathbf{h}(\hat{\mathbf{x}}_k) = \begin{bmatrix} \hat{\mathbf{q}}_k \\ \overset{+}{2} \hat{\mathbf{q}}_k \hat{\mathbf{q}}_k \end{bmatrix}. \quad (96)$$

Fig. 5 Scheme of the test platform

The equations above are suitable for the SRUKF/UKF observer, however, since the authors use also the SRUCKF observer, a modified state vector and output vector should be proposed. For any filter where $m = n$, the following can be adopted:

$$\hat{\mathbf{x}}_k = \begin{bmatrix} \hat{\mathbf{q}}_k^T & \hat{\mathbf{q}}_k^T & (\hat{\omega}_k^b)^T \end{bmatrix}^T, \quad (97)$$

$$\hat{\mathbf{y}}_k = \hat{\mathbf{x}}_k, \quad (98)$$

for which the model function (system model) is given as:

$$\mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}) = \begin{bmatrix} \hat{\mathbf{q}}_{k-1} + \frac{T}{2} \overset{+}{\hat{\mathbf{q}}}_{k-1} \mathbf{I}^{-1} \mathbf{s}_{k-1} - T \overset{+}{\hat{\mathbf{q}}} \left(\overset{+}{\hat{\mathbf{q}}}_{k-1} \right)^T \hat{\mathbf{q}}_{k-1} \\ \hat{\mathbf{q}}_{k-1} + T \overset{+}{\hat{\mathbf{q}}}_{k-1} \\ 2 \overset{+}{\hat{\mathbf{q}}}_{k-1} \hat{\mathbf{q}}_{k-1} \end{bmatrix}. \quad (99)$$

These two observers (SRUKF, SRUCKF) will be compared in terms of accuracy and robustness. Note that the state vectors are different, however, the aim is to find a precise tool for state estimation where only the attitude is important, so even after adding linear velocity or position the attention is still focused on orientation. The authors consider these two observers as possessing minimal complexity. Moreover, these observers are easily expandable with additional state elements, which makes them flexible.

5 State Estimation Results

In experiments the drone was sending the sensory data to the base station where also the data from the VICON motion capture system were gathered (Fig. 5). The VICON was used as the ground truth. The drone is a modified Crazyflie 2.0, for which BLDC motors were installed instead of the classic DC ones.

The research involves two main sources of information: the model and a low-cost AHRS (InvenSense MPU-9250), which is EKF-based (providing only quaternion and angular velocity, its main idea is given in [4]). The trail data gathered in the experiment allowed a basic AHRS statistic to be determined. The results are presented in Table 1. As it can be noticed, the estimation results are rather poor, however, they are sufficient for basic maneuvers, and a great majority

Table 1 AHRS-based estimate statistical parameters

Axis	Standard deviation	Value	Unit
X	$\sigma_x (\phi)$	3.46	(°)
Y	$\sigma_y (\theta)$	3.27	(°)
Z	$\sigma_z (\psi)$	3.31	(°)
	Mean error	Value	Unit
X	ϕ_e	2.42	(°)
Y	θ_e	2.16	(°)
Z	ψ_e	2.22	(°)
	Max error	Value	Unit
X	ϕ_m	17.4683	(°)
Y	θ_m	16.2383	(°)
Z	ψ_m	13.5089	(°)

of low-cost drones have similar AHRSs. This system is extended with additional observers which were described in the previous section. It is, however, important to introduce some basics of controllers and observers settings.

In the drone, six PID controllers are used, for which eighteen parameters have to be set. Observers, either the SRUKF or the SRUCKF, have 3 parameters to be set, i.e. α , β , and κ , which are set according to [9]. Additionally, observers have the covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$, where the elements on their main diagonals (variances) have to be set according to the state variables and measurements. In the case of the SRUKF observer, the elements of $\mathbf{Q}_k$, namely q_{ii} are as follows:

$$q_{ii} = \begin{cases} q_{\hat{q}}, & \text{for } i \in \langle 1 : 4 \rangle, \\ q_{\hat{v}}, & \text{for } i \in \langle 5 : 7 \rangle, \\ q_{\hat{q}}, & \text{for } i \in \langle 8 : 11 \rangle. \end{cases} \quad (100)$$

For the SRUCKF observer, q_{ii} are given in Eq. 101.

$$q_{ii} = \begin{cases} q_{\hat{q}}, & \text{for } i \in \langle 1 : 4 \rangle, \\ q_{\hat{q}}, & \text{for } i \in \langle 5 : 8 \rangle, \\ q_{\hat{\omega}}, & \text{for } i \in \langle 9 : 11 \rangle. \end{cases} \quad (101)$$

In the case of $\mathbf{R}_k$, its diagonal elements r_{ii} are equal for all i .

In the optimization process, the Particle Swarm Optimization (PSO) method is used, where the cost function is chosen so that the errors of attitude and position are taken into account:

$$C_f = \sum_{i=0}^l (|\hat{\mathbf{q}}_i^* \otimes \mathbf{q}_{set_i}| + |\mathbf{p}_{set_i} - \hat{\mathbf{p}}_i|). \quad (102)$$

The results of optimization of the SRUKF and the control settings are presented in Table 2, while the results of state estimation are given in Table 3.

If one compares the stand-alone AHRS with the SRUKF, it can be seen that the results are significantly in favor of

Table 2 Settings of the control algorithms and the state observer (SRUKF) for $n = 11$, $m = 7$

	x-p	y-p	z-p
K_p	0.278079	0.169538	418.202
K_i	1.00057e-010	1.10579e-009	0.165394
K_d	42.8906	23.9136	36749.7
	x-o	y-o	z-o
K_p	2834.84	1272.45	302.434
K_i	0	0	0
K_d	1241.84	1464.35	1278.67
$q_{\hat{q}}$	3.24319e-005		
$q_{\hat{v}}$	1.02934e-012		
$q_{\hat{q}}$	0.000157278		
r_{ii}	0.26647		
Cost (C_f)	729.953		

Table 3 Results of state estimation for SRUKF, $n = 11$, $m = 7$

Axis	Standard deviation	Value	Unit
X	$\sigma_x (\phi)$	2.1162	(°)
Y	$\sigma_y (\theta)$	1.9896	(°)
Z	$\sigma_z (\psi)$	2.3470	(°)
X	$\sigma_x (p_x)$	0.0152	(m)
Y	$\sigma_y (p_y)$	0.0306	(m)
Z	$\sigma_z (p_z)$	0.0339	(m)
	Mean error	Value	Unit
X	ϕ_e	1.3778	(°)
Y	θ_e	1.2588	(°)
Z	ψ_e	1.3876	(°)
X	p_{x_e}	0.0108	(m)
Y	p_{y_e}	0.0196	(m)
Z	p_{z_e}	0.0145	(m)
	Max error	Value	Unit
X	ϕ_m	16.6975	(°)
Y	θ_m	16.6894	(°)
Z	ψ_m	13.8183	(°)
X	p_{x_m}	0.0773	(m)
Y	p_{y_m}	0.1168	(m)
Z	p_{z_m}	0.3004	(m)

the SRUKF observer. The mean error has been reduced by at least one degree, while the standard deviation is lower for each axis. The maximum error is similar and could not be reduced. The trail data are presented in Fig. 6. As it can be seen, the AHRS estimate suffers from bias in Euler angles/ quaternions, which is significant. When these data are combined with the model prediction, it is possible to better imitate the drone's behavior. That was proved in the obtained results. Since the model was used in a closed loop with the observer in the feedback, the controller settings had to be optimized. The obtained optimization results are given in Table 2. As it can be noticed, the orientation controllers were simplified to the PD scheme, as the K_i parameters were small enough to be treated as zero.

In the next experiment the SRUCKF (the new observer), whose state function was given in Eq. 99, is studied. The process of controller settings tuning was repeated. By using the same cost function C_f , the results given in Table 4 were obtained. The SRUCKF estimation results are shown in Table 5. The orientation is estimated better if compared to the SRUKF, however, the cost is bigger, mainly because of the position, for which the set value could not be followed so well as in the case of the SRUKF. Since r_{ii} is lower than in the case of the SRUKF, the AHRS data are treated as less important in the estimation process. What is interesting, the integral parts of the position controllers have smaller

Fig. 6 Euler/Tait-Bryan angles - SRUKF estimation results for $n = 11, m = 7$

influence on the control process (Table 4). This stems from the observer construction, which does not have position and linear velocity included in the state vector. It also proves that the controller settings depend on the observer state structure, even when some of the state variables are not directly used in the control process. The trial data with the SRUCKF estimation results are given in Fig. 7.

These two observers provide a good estimate of orientation. The controller settings and the observer structures are different, however, the cost functions are similar. On the basis of the presented data it cannot be said which one would be better in a UAV application. Since the authors postulate fault-tolerant estimation in the paper,

Table 4 Settings of the control algorithms and the state observer (SRUCKF) for $n = 11, m = 11$

	x-p	y-p	z-p
K_p	0.309019	0.271551	83.1562
K_i	1.37555e-009	1.38898e-009	1.14674e-009
K_d	27.2932	26.2699	16244.6
	x-o	y-o	z-o
K_p	976.63	589.715	736.907
K_i	0	0	0
K_d	2097.61	1893.44	6056.9
$q_{\hat{q}}$	5.30313e-007		
$q_{\hat{q}}$	0.000168057		
$q_{\hat{\omega}}$	0.000389901		
r_{ii}	0.0224998		
Cost (C_f)	748.089		

it is desirable to test their robustness against errors. For this purpose, an AHRS failure will be simulated. Among many possible scenarios of AHRS faults/failures, only one will be studied. In most of the AHRSs, the process is driven by external interruptions. They are often chosen to

Table 5 Results of state estimation for SRUCKF, $n = 11, m = 11$

Axis	Standard deviation	Value	Unit
X	$\sigma_x (\phi)$	1.8625	(°)
Y	$\sigma_y (\theta)$	2.1581	(°)
Z	$\sigma_z (\psi)$	1.9562	(°)
X	$\sigma_x (p_x)$	0.0303	(m)
Y	$\sigma_y (p_y)$	0.0320	(m)
Z	$\sigma_z (p_z)$	0.0480	(m)
	Mean error	Value	Unit
X	ϕ_e	1.2492	(°)
Y	θ_e	1.4096	(°)
Z	ψ_e	0.8627	(°)
X	p_{x_e}	0.0199	(m)
Y	p_{y_e}	0.0182	(m)
Z	p_{z_e}	0.0172	(m)
	Max error	Value	Unit
X	ϕ_m	11.1513	(°)
Y	θ_m	15.3769	(°)
Z	ψ_m	18.5476	(°)
X	p_{x_m}	0.1874	(m)
Y	p_{y_m}	0.1933	(m)
Z	p_{z_m}	0.4674	(m)

Fig. 7 Euler/Tait-Bryan angles - SRUCKF estimation results for $n = 11, m = 11$

be based on a particular sensor. When such sensor does not respond, a timeout procedure is invoked, so the process can be handled without one or more sensor data values. Although these emergency procedures can cope with some temporary issues, they will eventually fail if sensor reset

does not help. Also, when the UAV is exposed to ultrasonic waves, the MEMS-based sensory system may get damaged permanently. In these two scenarios the AHRS estimate will be frozen, or will simply walk randomly until it reaches

Table 6 Results of state estimation for SRUCKF, disrupted AHRS data

Axis	Standard deviation	Value	Unit
X	$\sigma_x (\phi)$	2.0304	(°)
Y	$\sigma_y (\theta)$	1.9807	(°)
Z	$\sigma_z (\psi)$	2.5918	(°)
X	$\sigma_x (p_x)$	0.0152	(m)
Y	$\sigma_y (p_y)$	0.0328	(m)
Z	$\sigma_z (p_z)$	0.0366	(m)
Mean error			
X	ϕ_e	1.3086	(°)
Y	θ_e	1.2462	(°)
Z	ψ_e	1.5793	(°)
X	p_{x_e}	0.0108	(m)
Y	p_{y_e}	0.0218	(m)
Z	p_{z_e}	0.0159	(m)
Max error			
X	ϕ_m	16.6975	(°)
Y	θ_m	16.6894	(°)
Z	ψ_m	13.8183	(°)
X	p_{x_m}	0.0698	(m)
Y	p_{y_m}	0.1168	(m)
Z	p_{z_m}	0.3004	(m)

Table 7 Results of state estimation for SRUCKF, disrupted AHRS data

Axis	Standard deviation	Value	Unit
X	$\sigma_x (\phi)$	1.8966	(°)
Y	$\sigma_y (\theta)$	2.1611	(°)
Z	$\sigma_z (\psi)$	1.9596	(°)
X	$\sigma_x (p_x)$	0.0305	(m)
Y	$\sigma_y (p_y)$	0.0322	(m)
Z	$\sigma_z (p_z)$	0.0486	(m)
Mean error			
X	ϕ_e	1.2576	(°)
Y	θ_e	1.4064	(°)
Z	ψ_e	0.8665	(°)
X	p_{x_e}	0.0199	(m)
Y	p_{y_e}	0.0186	(m)
Z	p_{z_e}	0.0178	(m)
Max error			
X	ϕ_m	11.1513	(°)
Y	θ_m	15.3769	(°)
Z	ψ_m	18.5477	(°)
X	p_{x_m}	0.1887	(m)
Y	p_{y_m}	0.1933	(m)
Z	p_{z_m}	0.4674	(m)

Fig. 8 Euler/Tait-Bryan angles - SRUCKF estimation results for $n = 11, m = 11$ (AHRS failure case)

a constant value. Although many other possible AHRS faults/failures exist, these are the most common ones. In order to verify observer robustness, the AHRS was prepared to simulate a failure by sending a constant signal. The estimation results are given in Table 6, as well as in Table 7 and Fig. 8.

The result are similar to the previous case, when the sensory data were fault/failure-free. It means that, for the observer settings used, the algorithms immune to sensory data disruption. Both the SRUKF and SRUCKF have proved to be robust. This feature is, however, not inherited with a given observer. It occurs normally when the Kalman Filter settings strongly limit the trust in correction data, i.e. sensory data, which is clearly the case. It also implies that in the case when the sensory system is of industrial or tactical grade, this feature will not be so evident. For such systems adaptive observers have to be used. The robustness test reveals that the observers are equally good when applied in the UAV. It also shows that with the given framework (combining the ARHS and the model data) an FTO can be obtained, making it possible to tackle down sensory system failures or faults.

6 Conclusions

In the article, two observers have been introduced. The first one is based on the SRUKF, and the second one is based on the modified Kalman Filter called SRUCKF. For these two observers, different state functions were proposed. In all studied scenarios, the closed-loop scheme was used,

where the input was the set position. Thanks to the proposed control structure and the precise robot parameters it was possible to follow the set position similarly as it is done in a drone. The authors have focused on the orientation, which is normally estimated by the AHRS. By introducing the closed-loop model, another source of information could be exploited. As it was proved, both observers can operate on the basis of the model and AHRS data. Moreover, during emergency situations the observers can operate incessantly while being only fed with drone model data, without valid correction data - FTO.

One of the assumptions in the research concerns the algorithms complexity, or the computational burden. In the case of the proposed observers, the state vectors have the same dimension, however, the output vectors differ. In the SRUKF $m = 7$, while for the SRUCKF $m = 11$ ($\dim(\mathbf{y}_k) = m$). Since the inverse matrix is needed, whose complexity is in general $\mathcal{O}(m^3)$, the SRUCKF will clearly not be optimal. However, in the same operation, the matrix multiplication is done, for which a diagonal matrix is used instead of a symmetric one. On that basis one can notice some advantages of the SRUCKF over the SRUKF. According to [9], the additional optimization mechanisms can be added to SRUKF/SRUCKF. This refers mainly to the matrix inversion. Since in the Kalman gain equation the upper triangular matrix $\mathbf{S}_y$ is used, the standard procedure of matrix inversion can be replaced by backward substitution. This reduces the inversion complexity to $\mathcal{O}(m^2)$. After applying the backward substitution method, the SRUCKF can compete with the SRUKF.

In the paper, robustness of the presented methods was tested. As it was proved, both observers can cope with temporary sensory faults or even permanent failures. It is, however, important to stress the role of observer settings. In the experiment, a drone with a low-cost AHRS was used. It simply induced low trust for correction data, and thus it implied observer robustness. This, however, has to be changed when an industrial or tactical grade sensor is used. For such systems, an adaptive mechanism must be used, which will be subjected to future investigations.

Acknowledgments The authors would like to express their gratitude to Mr. Xiang He from the University of Utah, for successful cooperation in the field of robotics and for providing the necessary flight data for a number of studies.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Appendix A: Algorithms of Kalman Filter, Using UT

A.1 Unscented Kalman Filter

The Unscented Kalman Filter is as a multistep, recursive algorithm. In the first step the whole process is initialized. It is done by calculating the *a posteriori* state vector $\mathbf{x}_0$ and covariance matrix $\mathbf{P}_0$:

$$\hat{\mathbf{x}}_0 = \frac{1}{k} \sum_{i=0}^k \mathbf{x}_{0i}, \quad (103)$$

$$\mathbf{P}_0 = \frac{1}{k} \sum_{i=0}^k (\mathbf{x}_{0i} - \hat{\mathbf{x}}_0)(\mathbf{x}_{0i} - \hat{\mathbf{x}}_0)^T, \quad (104)$$

where i is in the range $< 1; k >$ and stands for the measurement sample number (in general: $k < 10$). This applies only when the state is measurable, otherwise the unobservable elements are set to zero. The UKF procedure can be presented in the following steps:

$$\chi_{k-1}^{a,i} = \hat{\mathbf{x}}_{k-1} + \text{sign}(n-i) \left(\sqrt{(n+\lambda)\mathbf{P}_{k-1}} \right)_i^T, \quad (105)$$

$$\chi_{k-1}^i = \mathbf{f}(\chi_{k-1}^{a,i}, \mathbf{u}_{k-1}), \quad (106)$$

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2n} W_i^{(m)} \chi_{k-1}^i, \quad (107)$$

$$\mathbf{P}_k^- = \sum_{i=0}^{2n} W_i^{(c)} \left(\chi_{k-1}^i - \hat{\mathbf{x}}_k^- \right) \left(\chi_{k-1}^i - \hat{\mathbf{x}}_k^- \right)^T + \mathbf{Q}_{k-1}, \quad (108)$$

$$\chi_k^{a,i} = \hat{\mathbf{x}}_k^- + \text{sign}(n-i) \left(\sqrt{(n+\lambda)\mathbf{P}_k^-} \right)_i^T, \quad (109)$$

$$\mathcal{Y}_k^i = \mathbf{h}(\chi_k^{a,i}), \quad (110)$$

$$\hat{\mathbf{y}}_k = \sum_{i=0}^{2n} W_i^{(m)} \mathcal{Y}_k^i, \quad (111)$$

$$\mathbf{P}_{xy} = \sum_{i=0}^{2n} W_i^{(c)} (\chi_k^i - \hat{\mathbf{x}}_k^-) (\mathcal{Y}_k^i - \hat{\mathbf{y}}_k)^T, \quad (112)$$

$$\mathbf{P}_{yy} = \sum_{i=0}^{2n} W_i^{(c)} (\mathcal{Y}_k^i - \hat{\mathbf{y}}_k) (\mathcal{Y}_k^i - \hat{\mathbf{y}}_k)^T + \mathbf{R}_k, \quad (113)$$

$$\mathbf{K}_k = \mathbf{P}_{xy} \mathbf{P}_{yy}^{-1}, \quad (114)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \hat{\mathbf{y}}_k^-), \quad (115)$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_{yy} \mathbf{K}_k^T \quad (116)$$

for $i \in < 1; 2n >$. Note that $\text{sign}(\dots)$ has an exception, i.e. $\text{sign}(0) = 0$.

Parameters are given as follows:

$$\lambda = (n + \kappa) \alpha^2 - n, \quad (117)$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(n + \lambda)} \quad i \in (< 0 : 2n > \setminus n), \quad (118)$$

$$W_n^{(m)} = \frac{\lambda}{2(n + \lambda)}, \quad (119)$$

$$W_n^{(c)} = \frac{\lambda}{2(n + \lambda)} + (1 - \alpha^2 + \beta), \quad (120)$$

where α , β , and κ are tuning parameters: α is responsible for sigma points spread, κ is a scaling parameter, and β is noise distribution parameter (set to 2 for Gaussian distributions).

The UKF has some significant differences when compared to the EKF or the KF. First of all, it does not approximate the covariance matrix $\mathbf{P}_k^-$ or $\mathbf{P}_k$, instead it transfers the so-called "sigma points" ($\mathcal{Y}_k^i$ and χ_k^i) through nonlinear functions ($\mathbf{f}(\dots)$ and $\mathbf{h}(\dots)$) and calculates weighted covariances from these points. It is a simple and fast method, provided that the sigma points are properly calculated. A single sigma point is a sum of the state vector (*a priori* or *a posteriori*) and a column or a row of the square root of a given covariance matrix (*a priori* or *a posteriori*, respectively).

A.2 Square Root Unscented Kalman Filter

The process is initialized with the *a posteriori* estimate and the square root of the *a posteriori* covariance matrix:

$$\hat{\mathbf{x}}_0 = \frac{1}{k} \sum_{i=0}^k \mathbf{x}_{0i}, \quad (121)$$

$$\mathbf{S}_0 = \text{chol} \left(\frac{1}{k} \sum_{i=0}^k (\mathbf{x}_{0i} - \hat{\mathbf{x}}_0)(\mathbf{x}_{0i} - \hat{\mathbf{x}}_0)^T \right). \quad (122)$$

The algorithm can have the following layout:

$$\chi_{k-1}^{a,i} = \hat{\mathbf{x}}_{k-1} + \text{sign}(n-i)\sqrt{(n+\lambda)} (\mathbf{S}_{k-1})_i^T, \quad (123)$$

$$\chi_{k-1}^i = \mathbf{f}(\chi_{k-1}^{a,i}, \mathbf{u}_{k-1}), \quad (124)$$

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2n} W_i^{(m)} \chi_{k-1}^i, \quad (125)$$

$$\mathbf{S}_k^- = \text{qr} \left\{ \left[\sqrt{W_i^{(c)}} \chi_{k-1}^i - \hat{\mathbf{x}}_k^- \quad \sqrt{Q_{k-1}} \right]^T \right\}, \quad (126)$$

$$i \in (< 0 : 2n > \setminus n), \quad (127)$$

$$\mathbf{S}_k^- = \text{cholupdate} \left\{ \mathbf{S}_k^-, \chi_{k-1}^n - \hat{\mathbf{x}}_k^-, W_n^{(c)} \right\}, \quad (128)$$

$$\chi_k^{a,i} = \hat{\mathbf{x}}_k^- + \text{sign}(n-i)\sqrt{(n+\lambda)} (\mathbf{S}_k^-)_i^T, \quad (129)$$

$$\mathcal{Y}_k^i = \mathbf{h}(\chi_k^{a,i}), \quad (130)$$

$$\hat{\mathbf{y}}_k = \sum_{i=0}^{2n} W_i^{(m)} \mathcal{Y}_k^i, \quad (131)$$

$$\mathbf{S}_y = \text{qr} \left\{ \left[\sqrt{W_i^{(c)}} ((\mathcal{Y}_k^i - \hat{\mathbf{y}}_k)) \quad \sqrt{R_k} \right]^T \right\}, \quad (132)$$

$$\text{where } i \in (< 0 : 2n > \setminus n), \quad (133)$$

$$\mathbf{S}_y = \text{cholupdate} \left\{ \mathbf{S}_y, \mathcal{Y}_k^n - \hat{\mathbf{y}}_k, W_n^{(c)} \right\}, \quad (134)$$

$$\mathbf{P}_{xy} = \sum_{i=0}^{2L} W_i^{(c)} \left[\chi_k^i - \hat{\mathbf{x}}_k^- \right] \left[\mathcal{Y}_k^i - \hat{\mathbf{y}}_k \right]^T, \quad (135)$$

$$\mathbf{K}_k = (\mathbf{P}_{xy} / \mathbf{S}_y^T) / \mathbf{S}_y, \quad (136)$$

$$\mathbf{x}_k = \mathbf{x}_k^- + \mathbf{K}_k (\mathbf{y}_k - \hat{\mathbf{y}}_k), \quad (137)$$

$$\mathbf{U} = \mathbf{K}_k \mathbf{S}_y, \quad (138)$$

$$\mathbf{S}_k = \text{cholupdate} \left\{ \mathbf{S}_k^-, \mathbf{U}_i, -1 \right\} \quad i \in (< 1 : m >). \quad (139)$$

Appendix B: Lemmas and Proofs

Lemma 1 *Diagonalizable matrix.*

If matrix $\mathbf{A}$ is diagonalizable, then there exists matrix $\mathbf{P}$, such as:

$$\mathbf{P}^{-1} \mathbf{A} \mathbf{P} = \mathbf{D}, \quad (140)$$

where $\mathbf{D}$ is diagonal.

Lemma 2 *Diagonalizable symmetric matrix.*

If matrix $\mathbf{A}$ is symmetric, then it is diagonalizable with orthogonal matrices, such as:

$$\mathbf{Q}^{-1} \mathbf{A} \mathbf{Q} = \mathbf{D}, \quad (141)$$

where $\mathbf{Q}$ is orthogonal matrix and $\mathbf{D}$ is diagonal.

Proof Square root of a symmetric matrix.

If matrix $\mathbf{A}$ is decomposed to diagonal and orthogonal matrix, i.e.

$$\mathbf{A} = \mathbf{Q} \mathbf{D} \mathbf{Q}^T, \quad (142)$$

then its square root $\mathbf{B}$ ($\mathbf{A} = \mathbf{B} \mathbf{B}$) is equal to the following:

$$\mathbf{B} = \mathbf{Q} \sqrt{\mathbf{D}} \mathbf{Q}^T, \quad (143)$$

which is proven by the following:

$$\begin{aligned} \mathbf{B} \mathbf{B} &= \mathbf{Q} \sqrt{\mathbf{D}} \mathbf{Q}^T \mathbf{Q} \sqrt{\mathbf{D}} \mathbf{Q}^T \\ &= \mathbf{Q} \sqrt{\mathbf{D}} \sqrt{\mathbf{D}} \mathbf{Q}^T \\ &= \mathbf{Q} \mathbf{D} \mathbf{Q}^T \\ &= \mathbf{A}. \end{aligned} \quad (144)$$

□

Lemma 3 *Derivative of a scalar with respect to a matrix.*

For any matrix $\mathbf{X} \in \mathbb{R}^{(n \times m)}$ and variable y (which is equal to a function of $\mathbf{X}$) it is true that:

$$\frac{\partial y}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial y}{\partial x_{11}} & \frac{\partial y}{\partial x_{12}} & \cdots & \frac{\partial y}{\partial x_{1m}} \\ \frac{\partial y}{\partial x_{21}} & \frac{\partial y}{\partial x_{22}} & \cdots & \frac{\partial y}{\partial x_{2m}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y}{\partial x_{n1}} & \frac{\partial y}{\partial x_{n2}} & \cdots & \frac{\partial y}{\partial x_{nm}} \end{bmatrix}. \quad (145)$$

On the basis of Eq. 145, matrix trace derivative properties can be proven. For any matrices $\mathbf{X}$, $\mathbf{Y}$ and symmetric matrix $\mathbf{Z}$ it can be written:

$$\frac{\partial \text{tr}(\mathbf{X} \mathbf{Y})}{\partial \mathbf{X}} = \mathbf{Y}^T, \quad (146)$$

$$\frac{\partial \text{tr}(\mathbf{X} \mathbf{Z} \mathbf{X}^T)}{\partial \mathbf{X}} = 2 \mathbf{X} \mathbf{Z}. \quad (147)$$

Lemma 4 *Kronecker product of quaternions in a matrix-based notation.*

Let us assume that three quaternions: α , β , and γ are given in the following relation:

$$\gamma = \alpha \otimes \beta. \quad (148)$$

By introducing the matrix forms of α and β it can be written that:

$$\gamma = \overset{+}{\alpha} \beta = \overset{-}{\beta} \alpha, \quad (149)$$

where:

$$\overset{+}{\alpha} = \begin{bmatrix} a_0 & -\mathbf{a}^T \\ \mathbf{a} & a_0 \mathbf{I} + \tilde{\mathbf{a}} \end{bmatrix}, \quad \overset{-}{\beta} = \begin{bmatrix} b_0 & -\mathbf{b}^T \\ \mathbf{b} & b_0 \mathbf{I} - \tilde{\mathbf{b}} \end{bmatrix}. \quad (150)$$

In Eq. 150, $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{b}}$ are skew symmetric matrices:

$$\tilde{\mathbf{a}} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}, \quad \tilde{\mathbf{b}} = \begin{bmatrix} 0 & -b_3 & b_2 \\ b_3 & 0 & -b_1 \\ -b_2 & b_1 & 0 \end{bmatrix}. \quad (151)$$

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

- Leunberger, D.: An introduction to observers. *IEEE Trans. Autom. Control* **16**(6), 596–602 (1971)
- Kalman, R.E.: A new approach to linear filtering and prediction problems. *Trans. ASME — J. Basic Eng.* **82**(D), 35–45 (1960)
- Gelb, A., Kasper, J.F., Nash, R.A., Price, R.A., Sutherland, A.A.: *Applied Optimal Estimation*. MIT Press, Massachusetts (1974)
- Gośliński, J., Nowicki, M., Skrzypczyński, P.: Performance comparison of EKF-based algorithms for orientation estimation on android platform. *IEEE Sensors J* **15**(7), 3781–3792 (2015)
- Speyer, J.L., Crues, E.Z.: On-line aircraft state and stability derivative estimation using the modified-gain extended Kalman filter. *J. Guid. Control* **10**(3), 262–268 (1987)
- Wendel, J., Maier, A., Metzger, J., Trommer, G.F.: Comparison of extended and sigma-point Kalman filters for tightly coupled GPS/INS integration. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pp. 1–15 (2005)
- Chowdharya, G., Jategaonkar, R.: Aerodynamic parameter estimation from flight data applying extended and unscented Kalman filter. *Aerosp. Sci. Technol.* **14**(2), 106–117 (2010)
- Julier, S.J., Uhlmann, J.K.: New extension of the Kalman filter to nonlinear systems. In: *Proc. of the Signal Processing, Sensor Fusion, and Target Recognition*, vol. 3068, pp. 182–193, Orlando (1997)
- Merwe, R., Wan, E.A.: The square-root unscented Kalman filter for state and parameter-estimation. In: *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, (ICASSP '01), vol. 6, pp. 3461–3464 (2001)
- Crassidis, J.L., Markley, F.L.: Unscented filtering for spacecraft attitude estimation. *J. Guid. Control, Dyn.* **26**(4), 536–542 (2003)
- St-Pierre, M., Gingras, D.: Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system. In: *Proc. of the IEEE Intelligent Vehicles Symposium*, pp. 831–835 (2004)
- Huang, H., Yang, M., Zang, W., Wu, S., Pang, Y.: In vitro identification of four-element Windkessel models based on iterated unscented Kalman filter. *IEEE Trans. Biomed. Eng.* **58**(9), 2672–2680 (2011)
- Francis, J.G.F.: The QR transformation, *I. Comput. J.* **4**(1), 265–271 (1961)
- Gośliński, J., Gardecki, S., Giernacki, W.: Introduction of the flying robots into the human environment: An adaptive square-root unscented Kalman filter for a fault tolerant state estimation in a quadrotor. In: *Proc. of the Int. Conf. on Intelligent Environments*, pp. 117–123, Shanghai (2014)
- Samy, I., Postlethwaite, I., Gu, D.: Survey and application of sensor fault detection and isolation schemes. *Control. Eng. Pract.* **19**, 658–674 (2011)
- Chen, F., Jiang, R., Zhang, K., Jiang, B., Tao, G.: Robust backstepping sliding mode control and observer-based fault estimation for a quadrotor UAV. *IEEE Trans. Ind. Electron.* **63**(8), 5044–5056 (2016)
- Zhang, Z., Meng, X., Wu, J.: Quaternion-based Kalman filter with vector selection for accurate orientation tracking. *IEEE Trans. Instrum. Meas.* **61**(10), 2817–2824 (2012)
- Aldrovandi, L., Hayajneh, M., Melega, M., Furci, M., Naldi, R., Marconi, L.: A smartphone based quadrotor: Attitude and position. In: *Proc. of the Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, pp. 1251–1259, Denver (2015)
- Rigatos, G.G.: Nonlinear Kalman Filters and Particle Filters for integrated navigation of unmanned aerial vehicles. *Robot. Auton. Syst.* **60**(1), 978–995 (2016)
- Marantos, P., Koveos, Y., Kyriakopoulos, K.J.: UAV State estimation using adaptive complementary filters. *IEEE Trans. Control Syst. Technol.* **24**(1), 1214–1226 (2016)
- Gośliński, J., Gardecki, S., Giernacki, W.: Unscented Kalman filter for an orientation module of a quadrotor mathematical model. In: *Proc. of the 9-th Asian Control Conference*, pp. 1–6, Istanbul (2013)
- Caliskan, F., Hajiyeve, C.: Active fault-tolerant control of UAV dynamics against sensor-actuator failures. *J. Aerosp. Eng.* **29**, 1–17 (2016)
- Merheb, A., Noura, H., Bateman, F.: Active fault tolerant control of quadrotor UAV using sliding mode control. In: *Proc. of the International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 156–166 (2014)
- Lu, D., Zeng, G., Liu, J.: Non-fragile simultaneous actuator and sensor fault-tolerant control design for Markovian jump system based on adaptive observer. *Asian J. Control* **20**, 125–134 (2018)
- Giernacki, W., Skwierczyński, M., Witwicki, W., Wroński, P., Kozierski, P.: Crazyflie 2.0 Quadrotor as a Platform for Research and Education in Robotics and Control Engineering. In: *Proc. of the 22nd International Conference on Methods and Models in Automation and Robotics*, pp. 37–42, Miedzyzdroje (2017)
- Baca, T., Spurny, V., Penicka, R., Stepan, P., Hert, D., Saska, M., Thomas, J., Loianno, G., Kumar, V.: Autonomous landing on a moving vehicle with an unmanned aerial vehicle. Article accepted to: *Journal of Field Robotics*, 1–26 (2018)
- Spurny, V., Baca, T., Saska, M., Penicka, R., Krajník, T., Thomas, J., Dinesh, T., Loianno, G., Kumar, V.: Cooperative autonomous search, grasping and delivering in a treasure hunt scenario by a team of UAVs. Article accepted to: *Journal of Field Robotics*, 1–34 (2018)
- Giernacki, W., Gośliński, J.: Coaxial quadrotor – from the most important physical aspects to a useful mathematical model for control purposes. In review (2018)
- Bangura, M., Mahony, R.: Nonlinear dynamic modeling for high performance control of a quadrotor. In: *Proc. of the Australasian conference on robotics and automation*, pp. 1–10, Wellington (2012)
- Wehage, R.A.: Quaternions and Euler parameters - a brief exposition. *Comput. Aided Anal. Optim. Mech. Syst. Dyn.* **9**, 47–180 (1984)
- Chou, J.C.K.: Quaternion kinematic and dynamic differential equations. *IEEE Trans. Robot. Autom.* **8**(1), 53–64 (1992)
- Fresk, E., Nikolakopoulos, G.: Full quaternion based attitude control for a quadrotor. In: *Proc. of the European Control Conference (ECC)*, pp. 3864–3869, Zurich (2013)

Jarosław Gośliński received the M.Sc. degree in control engineering and robotics from Poznan University of Technology (Poland) in 2011 and Ph.D. degree in 2016. His research interests include signal filtering and processing, MEMS sensors, state observers, nonlinear state estimation and identification, swarm robotics and artificial intelligence methods. He is a co-funder of the AISENS (<http://aisens.co/>), where he is currently working on intelligent sensors.

Wojciech Giernacki received the M.Sc. degree in control engineering and robotics from the Poznan University of Technology in 2006 and Ph.D. degree in 2011. Since 2011, he is an assistant professor at the Institute of Control, Robotics and Information Engineering of the same university. He founded and heads the PUT AeroLab and Unmanned Aerial Vehicles Research Group (<http://uav.put.poznan.pl/>). He was lecturing at the Shanghai University, Technical University of Madrid, Czech Technical University in Prague and Technical Institute in Braganca. He is an author or co-author of more than 60 publications in peer-reviewed international conferences, articles in impacted journals and 1 book. He serves as a reviewer for several impacted journals as well as a program committee member of relevant international conferences. His scientific interests are focused around the issues of UAVs, especially robust and adaptive control, optimization techniques, as well as data fusion from sensors.

Andrzej Królikowski graduated from the Poznan University of Technology (1968). He received the Ph.D. and D.Sc. degrees in control engineering and robotics from the same University in 1974 and 1995, respectively. Till 2017 he was a full professor at the Poznan University of Technology. He is an author or co-author of more than 100 publications (at international conferences, articles in impacted journals, and books). His research interests include control theory, adaptive systems and identification.