Proceeding of the 2015 IEEE
International Conference on Information and Automation
Lijiang, China, August 2015

Generalized complementary filter for attitude estimation based on vector observations and cross products*

Xiang LI and Chuan HE
School of Electronic Engineering and Automation
Guilin University of Electronic Technology
Guilin, Guangxi Province, China
xli1984@hotmail.com

Yongjun WANG and Zhi LI
Department of Electronic Engineering
Guilin University of Aerospace Technology
Guilin, Guangxi Province, China
cclizhi@guat.edu.cn

Abstract **- A generalized complementary filter (GCF) for attitude estimation is presented in this paper, which is based on vector observation and its cross product. With brief reviews of introductions and discussions of complementary filters in the existing literature, it is pointed out that the vector cross product plays a key role in the basis of complementary attitude filter. Both the estimation and compensation of attitude error is carried out by means of cross products. Numerical simulation and application test are performed to evaluate the proposed GCF. Simulation and experiment results show that the proposed GCF has better numerical stability and much higher computational efficiency than the multiplicative extended Kalman filter (MEKF).**

Index Terms - Attitude estimation, complementary filter, vector observation, cross product.

I. INTRODUCTION

Attitude estimation is a fundamental problem in design of mobile robot and unmanned aerial vehicle (UAV). It is also referred to as ‘attitude filtering’, since it usually includes the filtering of raw measurement data with noise. A variety of attitude estimators can be found in the existing literature [1]. However, in most applications, the workhorse of real-time attitude estimation is the extended Kalman Filter (EKF), due to its relatively low computational burden.

Another attractive attitude estimation technique is the complementary filter. Since it has much higher computational efficiency, complementary filter is particularly suitable for low-cost applications, such as inertial measurement unit (IMU) and attitude and heading reference system (AHRS) that consist of MEMS sensors.

The basic idea of the classical complementary filter is to reconstruct the signal of interest in the frequency domain, i.e. the filter is designed according to the frequency characteristics of the sensor noise, rather than its statistical properties in the time domain. Therefore, the design and tuning of complementary filter is much more straightforward than that of EKF. Nevertheless, a remarkable fact is that the complementary filter has some essential relationships with Kalman filter. In some sense, a complementary filter can be viewed as a fixed gain Kalman filter [2].

In the field of attitude estimation, a noteworthy aspect is the attitude representation. From the mathematical point of view, the rotation of a rigid body can be described with the special orthogonal group SO(3) [3]. As a result, the dimension of a complementary attitude filter should be at least 3. Implementations of complementary attitude filters based on Euler angles [4] [5], quaternion [6] [7], and direction cosine matrix (DCM) [3] [8] are commonly used. Besides, Batista et al. [9] proposed a sensor-based complementary filter that utilizes vector observations (e.g. gravity and geomagnetic vectors) instead of explicit attitude representations like Euler angles, quaternion, DCM, etc. They argued that this strategy of complementary filter can ensure globally asymptotically stabilization. But their implementation of the ‘complementary’ filter was actually built on Kalman filtering architecture. Then Benziane et al. [10] developed two types of vector-based complementary attitude filters, i.e. the direct form and the passive form, which closely follows the structure of Mahony’s complementary filter in [3].

The above mentioned complementary filters were applied successfully. Still, the basis of complementary attitude filters remains ambiguous to the majority of researchers, partly because it involves the mathematical descriptions of 3D rotations and cannot be simply interpreted as a filter in the frequency domain.

In this paper, we focus on the analysis and implementation of complementary attitude filter based on vector observations. In Section II, we first review the classical complementary filter and the existing complementary filters in the field of attitude estimation, as well as their relationships to the EKF, and then clarify several critical issues in filter design, especially the key role of the cross product of vectors. After that, we propose a generalized complementary filter (GCF) based on vector observations. Simulation results and application on a quad-rotor UAV are presented in Section III and IV, respectively, followed by conclusion remarks in Section V.

II. METHODOLOGY

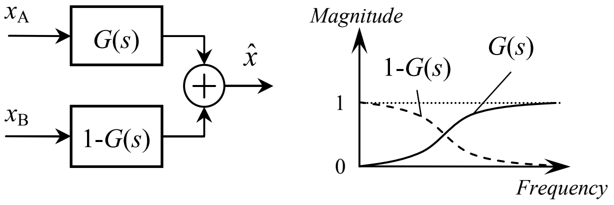
A. Review of the complementary filter

The basic complementary filter is illustrated in Fig.1, which produce the estimated signal $\hat{x}$ according to two independent measurements x_A and x_B . Since $G(s)$ is a high-

* This work is partially supported by the National Natural Science Foundation of China under Grant #61361006 to Zhi Li, and by the Guangxi Key Laboratory of Automatic Detecting Technology and Instruments under Grant #YQ14203 to Yongjun Wang and #YQ15107 to Xiang Li.

978-1-4673-9104-7/15/$31.00 ©2015 IEEE

pass filter, its complement $1-G(s)$ is a low-pass filter, and their combination will give an unbiased estimation $\hat{x}$.



(a)Block diagram (b)Transfer functions

Fig. 1 Basic complementary filter.

It should be noticed that the two measurements x_A and x_B in Fig.1 are both direct measurements of signal *x*. In the case of attitude filtering, however, we have miscellaneous inputs, which can be classified into two different categories. The first type directly includes the attitude information, e.g. the observations of gravity, geomagnetic field, and other vectors. The other kind is related to the changes of attitude, such as the measurements of angular rate or angular displacement provided by a gyroscope. Therefore, the complementary filter for attitude estimation should follow the architecture shown in Fig.2.

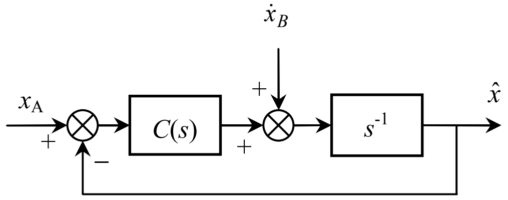


Fig.2 Block diagram of complementary attitude filter.

In Fig.2, x_A and $\dot{x}_B$ denotes the above two categories of measurements. The transfer function of Fig.2 is:

$$\hat{x} = \frac{C(s)\cdot x_A + s\cdot x_B}{C(s)+s}. \tag{1}$$

Different form of $C(s)$ can be applied in (1). For instance, in [3], [5] and [6], it was set to

$$C(s) = k_p + \frac{k_I}{s}, \tag{2}$$

which led to a second-order transfer function in (1). It is also possible to use more complicated forms of $C(s)$, in order to build higher order transfer functions [10].

B. Attitude error representation

In a practical attitude filter, the estimated signal $\hat{x}$ should be replaced by the attitude quaternion, DCM, or other attitude representations. Besides, the direct measurement x_A is usually displaced by one or more vector observations. Moreover, the time derivative of attitude will serve as $\dot{x}_B$. These three items cannot mathematically add or subtract each other. Thus the problem of representing attitude error arises, i.e. how to calculate $(x_A - \hat{x})$ in Fig.2.

In Mahony's original design of complementary attitude filter [3], the main architecture was based on DCM, and thus the attitude error was calculated using matrices multiplication. Since the dimension of DCM is too high (9 elements to be estimated), this approach is unsuitable for practical implementations, and it was soon replaced by the use of cross product of vectors [6].

As stated above, vector observations usually serve as the direct measurement x_A in most cases, and hence the cross product of vector observation and corresponding estimation is the most appropriate means to calculate and represent the attitude error. The representation of attitude error using vectors' cross product has been widely adopted in existing literature [5] [8] [10]. Furthermore, this approach is especially suitable to be applied together with the sensor-based filter in [9] and [10], since this kind of complementary filter is directly based on the vector observations.

Once we utilize the cross product to represent the attitude error, the implementation of complementary attitude filter can be summarized as follows. Assume that we have m independent vector observations that incorporate the attitude information. Let $\boldsymbol{v}_n^*$ denotes the nth vector's observation, and $\hat{\boldsymbol{v}}_n$ is the corresponding estimation. Besides, $\boldsymbol{\omega}^*$ denotes the angular velocity measured by the gyro. All the above vectors are represented in the carrier's body frame. Then we get the following expressions:

$$\boldsymbol{\omega}_{err} = \sum_{n=1}^{m} k_n \boldsymbol{v}_n^* \times \hat{\boldsymbol{v}}_n \tag{3}$$

$$\dot{\hat{\boldsymbol{v}}}_n = \hat{\boldsymbol{v}}_n \times \left[\boldsymbol{\omega}^* + \boldsymbol{C}(s)\boldsymbol{\omega}_{err}\right] \tag{4}$$

in which $\boldsymbol{C}(s)$ is a certain transfer function in 3×3 matrix form, and k_n is a positive weighting factor for the nth vector.

The filter architecture defined by (3) and (4) will be referred to as the generalized complementary filter (GCF) in the following discussions. Note that the GCF is a bit different from the filter proposed in [10]. This difference will be discussed in the following subsection.

C. The key roles of cross product

Both (3) and (4) include the cross product of the vector observation $\hat{\boldsymbol{v}}_n$. Actually, the cross product plays a critical role in the GCF. As it is well known, for a vector $\boldsymbol{u}$ that is fixed in the reference frame (or inertial frame, according to the inertial navigation theory), its time derivative $\dot{\boldsymbol{u}}$ in the carrier's body frame can be simply given by the following cross product:

$$\dot{\boldsymbol{u}} = \boldsymbol{u} \times \boldsymbol{\omega} \tag{5}$$

in which $\boldsymbol{\omega}$ is the carrier's angular velocity, represented in the body frame.

An important corollary of (5) is that the change of vector $\boldsymbol{u}$ caused by an infinitesimal 3D rotation during the time period Δt can be described by a cross product of the vector itself and an 'equivalent' angular velocity $\boldsymbol{\omega}_{eqv}$, as shown in (6).

$$\Delta\boldsymbol{u} \approx \left(\boldsymbol{u}\times\boldsymbol{\omega}_{eqv}\right)\Delta t \tag{6}$$

It must be pointed out that only the component of $\boldsymbol{\omega}_{eqv}$ that perpendicular to $\boldsymbol{u}$ will give rise to the change of $\boldsymbol{u}$, according to the property of cross product.

We then construct a triple vector product based on (6):

$$\boldsymbol{u}\times\Delta\boldsymbol{u} \approx \left(\boldsymbol{u}\times\boldsymbol{u}\times\boldsymbol{\omega}_{eqv}\right)\Delta t = \left|\boldsymbol{u}\right|^2\boldsymbol{\omega}_{eqv}\cdot\Delta t \tag{7}$$

Next, we denote the initial and final value of $\boldsymbol{u}$ as $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ respectively, then (7) can be rewritten as (8):

$$\boldsymbol{u}\times\Delta\boldsymbol{u} = \boldsymbol{u}_1\times\left(\boldsymbol{u}_2-\boldsymbol{u}_1\right) = \boldsymbol{u}_1\times\boldsymbol{u}_2 = \left|\boldsymbol{u}\right|^2\boldsymbol{\omega}_{eqv}\cdot\Delta t \tag{8}$$

According to (8), if a slight rotation occurs, we can get an equivalent angular velocity for this rotation by means of cross multiplying the initial and final observations of a vector.

Now we can clearly explain the intension of (3) and (4), which defines the architecture of GCF. The cross product $\boldsymbol{v}_n^*\times\hat{\boldsymbol{v}}_n$ in the right side of (3) indicates the difference between the n th vector observation and the corresponding estimation, and this cross product actually gives an 'equivalent' angular displacement. Then a correction is made to the gyro reading $\boldsymbol{\omega}^*$, which will compensate the attitude error.

We ask the reader to keep in mind that the attitude error is actually a 3D rotation, and thus it should be represented by the cross product as mentioned above. Moreover, the revision of attitude error should also be carried out via the cross product. This is the key point of GCF. In a sense, the GCF proposed in this paper is based on both the vector observations and their cross products.

D. More remarks

The time derivative of a vector's observation can be expressed as the cross product in (5). Coincidentally, the time derivatives of the attitude quaternion and DCM are also calculated by multiplication for quaternion and matrices, respectively. In a word, the attitude, as well as its error, is essentially multiplicative, rather than additive. So the dispute between 'additive' and 'multiplicative' attitude filter is insignificant.

However, the complementary filter proposed in [10] utilized both the additive and multiplicative attitude errors. The gyro bias was estimated by cross multiplying the vector observations and their estimations, and corrections to the estimated vectors were determined by subtract the estimations from the observations. That makes the filter complicated and unnatural.

Since the GCF discussed in the above subsections is based on the multiplicative attitude error, it can be naturally linked to the multiplicative EKF (MEKF). The internal relationship between complementary filter and MEKF has already been well stated by Jensen in [8].

A noteworthy aspect is that the cross product of a vector observation and its estimation can only give a projection of the attitude error on the orthogonal direction of the vector, rather than the entire attitude error. Therefore, we need at least two unparallel vectors to reconstruct the entire attitude error. This is the sufficient condition suggested by Jensen [8] and Benziane [10] for a complementary attitude filter to be asymptotically stable. But coincidently, for a certain vector, only the orthogonal component of attitude error to this vector will result in an error of its estimation. So we may deduce intuitively, that the GCF can remain stable even when only one vector observation is valid, although it will not give an complete estimation of the attitude error in this case. This conjecture awaits rigorous proof.

III. Numerical Simulation

A. Presumptions for simulation

In this section, we test the performance of the proposed GCF by numerical simulation, and compare it with MEKF and unscented Kalman filter (UKF). We use the following settings in the simulation:

1) The north-east-down (NED) coordinate frame is used as the reference frame, while the front-left-down coordinate frame of the carrier is chosen as the body frame. The gravity and geomagnetic vectors are set to be $\boldsymbol{h}_0$=(40 0 30)$^\mathrm{T}$ μT and $\boldsymbol{g}_0$=(0 0 9.8)$^\mathrm{T}$ m/s^2 respectively in the reference frame.
2) The standard derivation of the sensor noise is set to be: $\boldsymbol{\sigma}_m$=0.1μT for the magnetometer, $\boldsymbol{\sigma}_a$=0.01m/s^2 for the accelerometer, and $\boldsymbol{\sigma}_\omega$=0.05°/s for the gyro. Besides, the gyro bias is modeled as a first-order random walk process. Its initial value is a zero vector, and is driven by Gaussian white noise with standard derivation $\boldsymbol{\sigma}_b$=0.05°/s. The sampling rate of each sensor is 100Hz.
3) The body frame and the reference frame coincide with each other at the starting point, and then the carrier performs the 24 rotations in Table I successively. During each rotation, the angular rate increases linearly to 90°/s within 1s, and then decreases linearly to zero within 1s. Thus each rotation takes 2s and the rotation range is 90°.

TABLE I
Rotation sequence

Sequence no.	Starting & ending attitudes (directions of the body frame's *x*-, *y*-, and *z*-axis) [a]	Direction of the angular velocity
1	NED → UEN	+*y*
2	UEN → SEU	+*y*
3	SEU → DES	+*y*
4	DES → NED	+*y*
5	NED → DES	-*y*
6	DES → SEU	-*y*
7	SEU → UEN	-*y*
8	UEN → NED	-*y*
9	NED → NUE	-*x*
10	NUE → USE	+*z*
11	USE → SDE	+*z*
12	SDE → DNE	+*z*
13	DNE → NUE	+*z*
14	NUE → DNE	-*z*
15	DNE → SDE	-*z*
16	SDE → USE	-*z*
17	USE → NUE	-*z*
18	NUE → NWU	-*x*
19	NWU → NDW	-*x*

20	NDW → NED	-x
21	NED → NDW	+x
22	NDW → NWU	+x
23	NWU → NUE	+x
24	NUE → NED	+x

[a]N=north, E=east, W=west, S=south, U=upward, D=down.

B. Filter construction

The GCF is implemented according to (3) and (4), and the transfer function $C(s)$ in (2) is adopted. We set k_P=0.5, k_I=0.1, as well as k_n=0.5 for both the gravity and geomagnetic vectors.

On the other hand, we also use MEKF and UKF in the simulation, and make a comparison between the three types of attitude filters. Both the MEKF and UKF used in Section III are based on the multiplicative attitude error introduced by Markley [11], and it is denoted as $\boldsymbol{a}$. The state vector consists of the estimations of $\boldsymbol{a}$ and the gyro bias drift $\delta\boldsymbol{b}$. The kinematics equation can be described as

$$\begin{pmatrix}\dot{\hat{\boldsymbol{a}}}\\ \dot{\hat{\delta\boldsymbol{b}}}\end{pmatrix}=\begin{pmatrix}-[\boldsymbol{\omega}_c]_\times & \boldsymbol{I}_{3\times3}\\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3}\end{pmatrix}\begin{pmatrix}\hat{\boldsymbol{a}}\\ \hat{\delta\boldsymbol{b}}\end{pmatrix}+\begin{pmatrix}\boldsymbol{w}_a\\ \boldsymbol{w}_b\end{pmatrix}. \tag{9}$$

In (9), $\boldsymbol{\omega}_c$ is the corrected angular velocity (excluded the gyro bias), and $[\boldsymbol{\omega}_c]_\times$ is a 3×3 matrix that satisfied $[\boldsymbol{\omega}_c]_\times\boldsymbol{a}$ $=\boldsymbol{\omega}_c\times\boldsymbol{a}$. $\boldsymbol{w}_a$ and $\boldsymbol{w}_b$ are the process noise for $\boldsymbol{a}$ and $\delta\boldsymbol{b}$. Meanwhile, we use the errors of $\boldsymbol{g}$ and $\boldsymbol{h}$ (denoted as $\delta\boldsymbol{g}$ and $\delta\boldsymbol{h}$ respectively) to form the measurement vector for MEKF. The measurement equation is

$$\begin{pmatrix}\delta\boldsymbol{g}\\ \delta\boldsymbol{h}\end{pmatrix}=\begin{pmatrix}[\boldsymbol{g}^*]_\times & \boldsymbol{0}_{3\times3}\\ [\boldsymbol{h}^*]_\times & \boldsymbol{0}_{3\times3}\end{pmatrix}\begin{pmatrix}\boldsymbol{a}\\ \delta\boldsymbol{b}\end{pmatrix}+\begin{pmatrix}\boldsymbol{v}_g\\ \boldsymbol{v}_h\end{pmatrix}. \tag{10}$$

More details of the implementations of MEKF and UKF can be found in [11] and [12].

C. Results on different platforms

We use the directional errors of $\boldsymbol{g}$ and $\boldsymbol{h}$ to evaluate the filters' performance, which defined as the intersection angle between the estimation and the true value.

Firstly, we run the filtering program in Scilab, which is open-source software for numerical computation. Simulation results of GCF, MEKF, and UKF are shown in Fig. 3 to 5, as well as Table II.

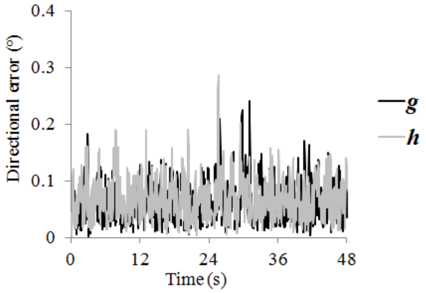


Fig.3. Simulation result of GCF.

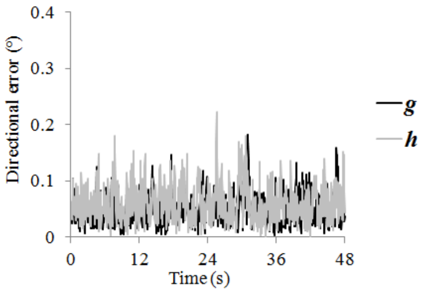


Fig.4. Simulation result of MEKF.

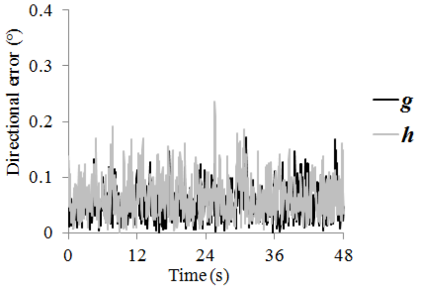


Fig.5. Simulation result of UKF.

TABLE II
SIMULATION RESULTS IN SCILAB

Filter type	GCF	MEKF	UKF
Average directional error of $\boldsymbol{g}$	0.0673°	0.0532°	0.0530°
Average directional error of $\boldsymbol{h}$	0.0695°	0.0658°	0.0657°
Total computing time	0.651s	0.908s	9.763s

It can be seen from Fig.3 to Fig.5 that the performance of GCF, MEKF, and UKF are quite similar to each other. According to Table I, the average errors of UKF and MEKF are nearly the same, but the former brings much higher computational burden.

Next, we run the filtering program on STM32F103RBT6. It is a 32-bit micro control unit (MCU) that based on Cortex-M3 core and manufactured by ST semiconductor. Simulation results are listed in Table III.

TABLE III
SIMULATION RESULTS ON 32-BIT MCU

Filter type	GCF	MEKF	UKF
Average error of $\boldsymbol{g}$	0.0673°	0.0608°	Diverging
Average error of $\boldsymbol{h}$	0.0695°	0.0797°	Diverging
Average computing time (in clock cycles)	1.531×10^4	3.206×10^5	–
Average computing time (with 72MHz system clock)	0.213ms	4.453ms	–

As shown in Table III, the average error of GCF remains the same as that in Scilab, while the error of MEKF shows an obvious increase. This is mainly because the computations on MCU are carried out with single precision floating point format, while Scilab uses double precision computations. This difference affects the accuracy of MEKF, but has no impact on GCF. Thus we can conclude that GCF has better numerical stability. On the other hand, a divergence happens to the UKF, which indicates that the UKF is not fit for running on an MCU.

The time consumption is also listed in Table III, which shows that the computational efficiency of GCF is much higher than that of MEKF. For a 32-bit MCU using 72MHz system clock, GCF can complete the update of both vectors within 0.213ms on average, while the average computing time of MEKF is up to 4.453ms. It is clear that the GCF can significantly ease the burden of MCU when it is applied to an AHRS.

IV. Application Test

The proposed GCF is applied to a quad-rotor UAV shown in Fig.6. The AHRS incorporated in this UAV consists of MPU6050 (3-axis accelerometer and 3-axis MEMS gyro) and HMC5883 (3-axis magnetometer), both of which have digital interface for data outputs.

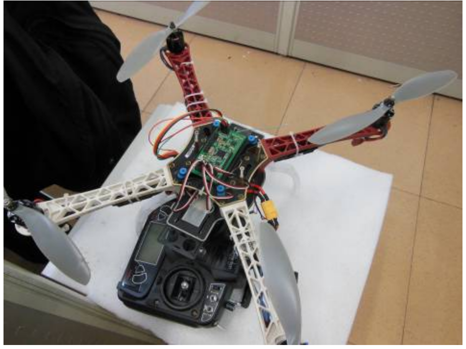


Fig.6. Quad-rotor UAV.

It is worth mentioning that MPU6050 contains an on-chip digital motion processor (DMP), which includes embedded motion fusion algorithms. Thus we evaluate our GCF by comparing its output with that of DMP. Test results are given in Fig.7 and Fig.8. This test is carried out under static condition, so that the pitch and roll angle can be precisely measured by an inclinometer for reference.

As shown in Fig.7 and Fig.8, the performance of GCF is similar to that of DMP. But the GCF can be easily tuned by the end user to achieve better adaptability.

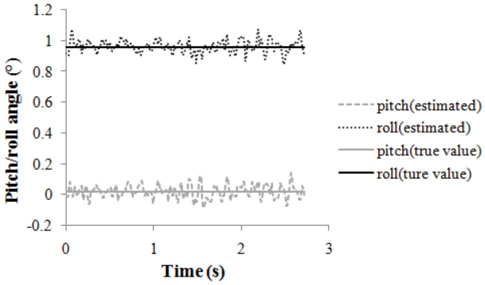


Fig.7. Pitch and roll outputs of GCF.

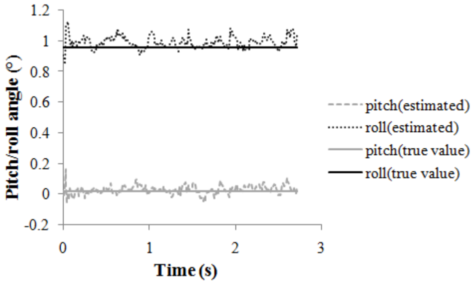


Fig.8. Pitch and roll outputs of DMP.

V. Conclusion

In this paper, we discuss the fundamentals of complementary attitude filter, and reveal that the vector cross product is the key to estimate and compensate the attitude error. Then we introduce a generalized complementary filter, and evaluate its performance by simulation and experiment. The proposed GCF shows ideal numerical stability and computational efficiency, and it is especially suitable to be executed on an MCU.

References

[1] J. L. Crassidis, F. L. Markley, and Y. Cheng, “Survey of nonlinear attitude estimation methods,” *J. Guid. Control. Dyn.*, vol. 30, no. 1, pp. 12-28, January–February 2007.

[2] W. T. Higgins Jr., “ A comparison of complementary and Kalman filtering,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 11, no. 3, pp. 321–325, May 1975.

[3] R. Mahony, T. Hamel, and J. M. Pflimlin, “Nonlinear complementary filters on the special orthogonal group,” *IEEE Trans. Autom. Control.*, vol. 53, no. 5, pp. 1203-1218, June 2008.

[4] J. F. Vasconcelos, B. Cardeira, C. Silvestre, P. Oliveira, and P. Batista, “Discrete-time complementary filters for attitude and position estimation: Design, analysis and experimental validation,” *IEEE Trans. Control. Syst. Technol.*, vol. 19, no. 1, pp. 181-198, January 2011.

[5] M. Blachuta, R. Grygiel, R. Czyba, and G. Szafranski, “Attitude and heading reference system based on 3D complementary filter,” in *2014 19th International Conference on Methods and Models in Automation and Robotics, MMAR 2014, Miedzyzdroje, Poland*, September 2-5 2014, pp. 851-856.

[6] R. Mahony, M. Euston, J. Kim, P. Coote, and T. Hamel, “A non-linear observer for attitude estimation of a fixed-wing unmanned aerial vehicle without GPS measurements,” *Trans. Inst. Meas. Control.*, vol. 33, no. 6, pp. 699-717, August 2011.

[7] V. Kubelka and M. Reinstein, “Complementary filtering approach to orientation estimation using inertial sensors only,” in *2012 IEEE International Conference on Robotics and Automation, ICRA 2012, Saint Paul, Minnesota, USA*, May 14-18 2012, pp. 599-605.

[8] K. J. Jensen, “Generalized nonlinear complementary attitude filter,” *J. Guid. Control. Dyn.*, vol. 34, no. 5, pp. 1588-1592, September-October 2011.

[9] P. Batista, C. Silvestre, and P. Oliveira, “Sensor-based globally asymptotically stable filters for attitude estimation: analysis, design, and performance evaluation,” *IEEE Trans. Autom. Control*, vol.57, no. 8, pp. 2095-2100, August 2012.

[10] L. Benziane, A. Benallegue and A. El Hadri, “A globally asymptotic attitude estimation using complementary filtering,” in *2012 IEEE International Conference on Robotics and Biomimetics, ROBIO 2012 - Conference Digest, Guangzhou, China*, December 11-14 2012, pp. 878-883.

[11] F. L. Markley, “Attitude error representations for Kalman filtering,” *J. Guid. Control Dyn.*, vol. 26, no. 2, pp. 311-317, March/April 2003.

[12] J. L. Crassidis and F. L. Markley, “Unscented filtering for spacecraft attitude estimation,” *J. Guid. Control Dyn.*, vol. 26, no. 4, pp. 536-542, July/August 2003.