Comparison of Complementary Filters Implementations for Unmanned Aerial Vehicles

Spyros Kontelis and Costas Psychalinos
Physics Department, Electronics Laboratory
University of Patras
Rio Patras, Greece
spyroskontelis@gmail.com; cpsychal@upatras.gr

***Abstract*—This work deals with the comparison between different analog implementations of complementary filters with a digital processing algorithm. For this purpose, different configurations of analog filters are tested in order to possibly find a better solution than that offered by the algorithm, used in many low-cost applications of Unmanned Aerial Vehicle positional awareness.**

***Index Terms*—Analog signal processing, complementary filters, unmanned aerial vehicles, gyroscope, accelerometer.**

I. Introduction

An Unmanned Aerial Vehicle (UAV) uses many sensors to correctly perform its mission. In order to achieve adequate positional awareness, at least, a 3-axis gyroscope and a 3-axis accelerometer must be employed. These sensors provide the signals, with which the onboard computer calculates the current position and orientation of the UAV. Each of these sensors has its own characteristics. The gyroscope produces an accurate signal for the estimation of angle of rotation between short intervals. It cannot, however, be reliable for lengthy periods of time, due to the drift of the signal. On the other hand, the accelerometer produces a signal without any drift, because of the gravitational field of the earth, but with a lot of high frequency noise, so it is not accurate enough for short/medium time interval estimations of change in orientation. In order to produce a signal with high accuracy, the complementary filtering technique is utilized [1]–[9]. This combines the estimation of each sensor, through a process (analog or digital) and produces a more reliable approximation, attenuating the noise and artifacts induced by each of the sensors [6], [10], [11].

A systematic comparison of possible implementations of of complimentary filters used in UAV applications is presented in this work. The implementations under consideration are discussed in detail in Section II, while their performance is evaluated in Section III.

II. Implementations of Complementary Filters

A. General setup

Let us consider the block diagram in Fig. 1. As a measuring device, the GY521 board with a MPU6050 IC is used, which combines a 3-axis gyroscope and a 3-axis accelerometer. The x, y, z axes correspond to the pitch, roll and yaw motion, respectively. It uses 6 and 16-bit ADCs for digitizing the gyroscope and the accelerometer outputs. The gyroscope measures the change in inclination from the previous moment of measurement and the accelerometer measures the change in acceleration in the same time interval. This is the Inertial Measurement Unit (IMU), which communicates with an Arduino Mega board via USB. For the encapsulation of the data a motion routine is created and is captured on video for later reference. The motion routine was done in a way that didn't exceed the measuring limits of the board, regarding the maximum measurable angular velocity and acceleration per second. After that, the Arduino processes the raw data in order to produce estimated angles and, in the case of Digital Filtering (DF), to calculate the filtered values. As a next step, it transfers every angle of rotation with its associated timestamp on a Personal Computer (PC), which handles the simulation of the analog filters and creates a 3D visualization of the results of each process (i.e., filtered and unfiltered). The resulting animation is used to visually compare the characteristics of each process and to select, without lengthy analysis, the process (i.e., digital or analog) and settings needed for the specific situation. The reason for this animation is based in the fact that, for N different settings of the filters, $3N$ angle-time plots are created and this is an extremely large number of 2D plots to be individually compared.

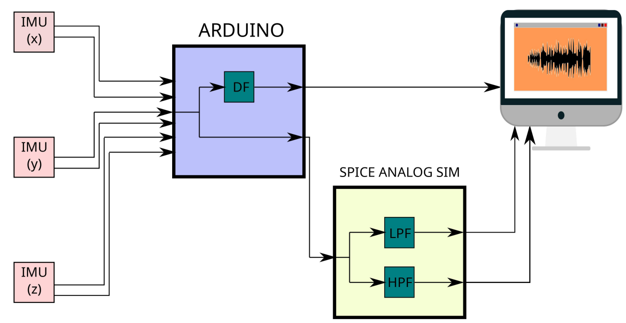


Fig. 1. Data processing path for handling the captured data and creating an animated comparison.

979-8-3503-9958-5/22/$31.00 ©2022 IEEE

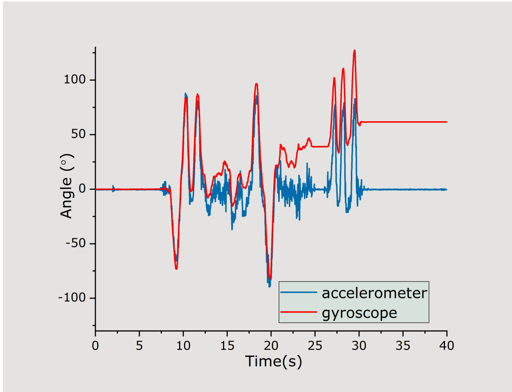


Fig. 2. Noisy signals given from accelerometer and gyroscope of the x–axis.

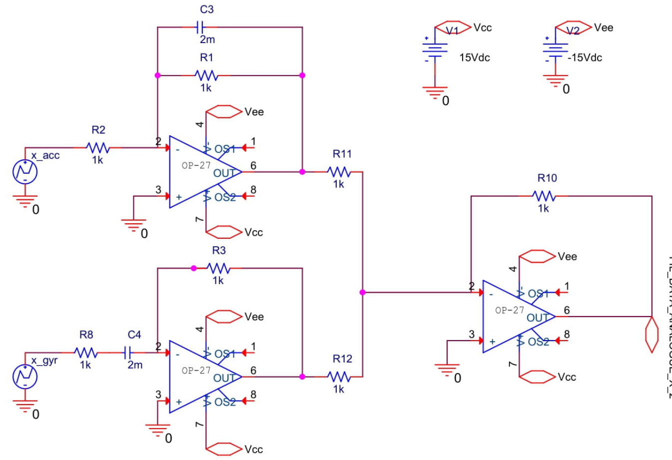


Fig. 3. 1st-order Butterworth complementary filter topology.The first stage creates the LPF and HPF and the second stage adds them.

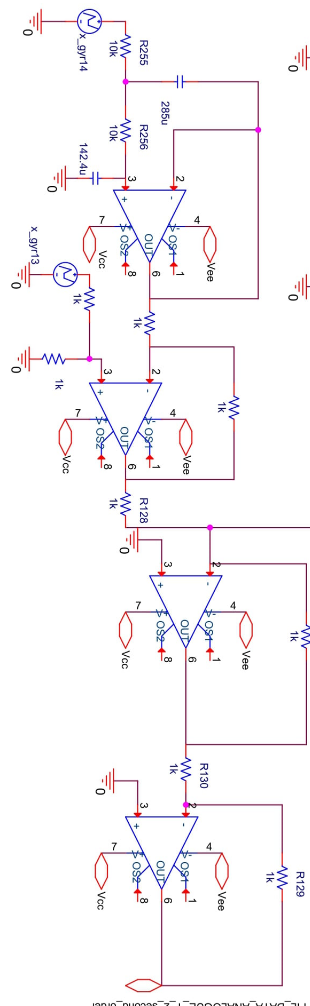


Fig. 4. 2nd–order Butterworth complementary filter topology.The HPF is created through the differentiation from the unit signal. After the summation of both filtered signals there is the topology of an inverter.

B. Analog filtering

When the unfiltered signals are plotted, the gyroscope data seem to have a (low frequency) DC component, because they drift in one direction all the time. It seems that the added error is integrated only when the sensor is moving and not when it is stationary. On the contrary, the accelerometer data seem to flick around an average value, showing a high frequency noise behavior as shown in Fig. 2. To mitigate these facts, the gyroscope data are processed through a high pass filter (HPF) and the accelerometer data through a low pass filter (LPF) [12], [13]. For comparison purposes, three differently tuned 1st–order Butterworth and three, similarly tuned in respect of cutoff frequency, 2nd–order Butterworth complementary filters based on the Sallen-Key implementation [14], have been utilized. The cutoff frequencies (f_c) were 79mHz, 31.8mHz and 15.9mHz, which correspond to time-constants (τ) equal to $2, 5, 10s$ respectively, for a 1st–order filter. These filters were used together with the differentiation as well as the summation topology, as it is shown in Figs. 3–4.

Each of the complementary filter designs was used three times, one for each axis of rotation, in order to create a 3D rotating block which emulates the roll and pitch motion of the real sensor-board unit. Reliable yaw motion could not be achieved with our sensor module, so we omitted it. In order to scale and convert the IMU voltages to degrees of rotation, the gain of the filters was 10.2 in all implementations. This value was obtained through the comparison of the output of the ADCs with the rotation of the module in our experimental setup. The 0V should correspond to 0° and the maximum value of the accelerometer should correspond to 90°. All of the filters where designed in OrCAD PSpice simulator, using the OP-27 operational amplifier discrete component IC.

C. Digital processing for filtering the signals

When the Arduino collects the data from the sensor, it converts them into angles. The degrees per second measured by the gyroscope are integrated over the time span and the accelerometer. As it is mentioned in Fig. 5, the roll angle is integrated to the previous cumulative roll angle, then, taking 96% of it and adding 4% of the calculated acceleration angle,

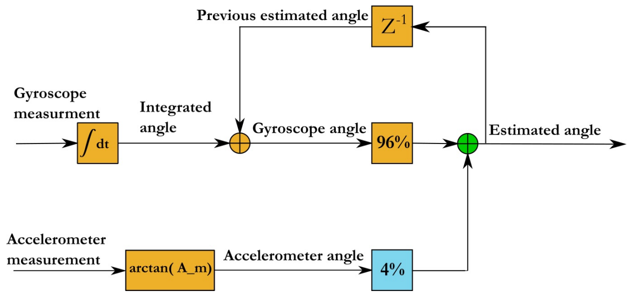


Fig. 5. Digital processing path for filtering the gyroscope and accelerometer.

in order to produce the software processed estimation. All the filtered and unfiltered data are forwarded to the computer for further processing. When all the data will be passed through the Arduino and OrCAD PSpice, they will be forwarded in the Processing integrated development environment (IDE) and will be displayed as cuboids that rotate, in order to be compared with the actual footage taken when the raw data were captured. Also, a hardware circuit was made in order to verify that the simulated analog filters were performing as expected. This is demonstrated in Fig. 6.

The source code material for the above is available in https://github.com/konteliss/Complementary-Filters-Comparison/tree/master.

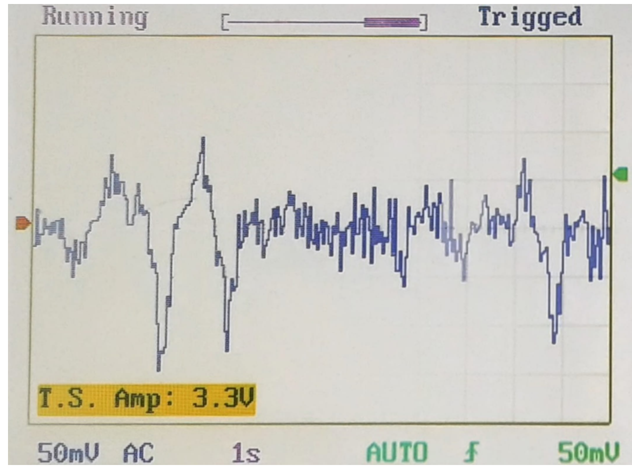
III. Comparison Results

A. Calculating the position

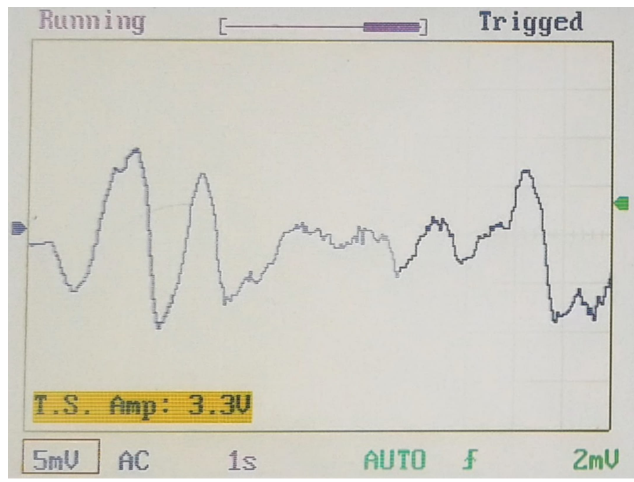
When comparing the simulated analog filters with the hardware realized one, it was clear that the simulation was up to par. This result indicates that the simulated analog filters produce reliable results and that they could be trusted for the full data set that they produced. Considering simulated analog filters and digitally processed data, the simulated filters had different characteristics depending on the order. In Fig. 7, each block represents the rotation of the sensor, as measured from the gyro (red) or the accelerometer (blue) or the calculated values given from each processing path (i.e. digital processing or analog filtering). This Figure depicts the reported rotation the moment the sensor was made to have zero rotation on each axis.

For comparison purposes, the values of the x and y axis of rotation of each block for this time-stamp and two more, without rotation, are given in TableI. No-one of the simulated analog filter achieves the performance of the software processed data, and this could be explained by the values picked for the cutoff frequencies. The trend was that higher cutoff frequencies resulted in a better approximation and none of the higher order filters we used attenuated the high frequency noise as the 1st–order did.

From these results the software combination should be picked as the optimal solution for positional awareness. To improve the results of the analog filters, different cutoff frequencies should be considered and simulated in order to



(a)



(b)

Fig. 6. (a) Input and (b) Output of 1st–order low pass filter, with τ=2s.

eliminate the long relaxation time and improve the accuracy of the system.

B. Comparing power requirements

The evaluation we made previously, was made only based on the resulting position. No other factor was taken into account. From the data-sheets of the Arduino and the simulated filters we measured the required power for producing each result. For each operational amplifier used in the analog filters there is a need for 100mW [15]. Consequently, for the three 1st–order complementary filters there is a need of 900mW of power. For the 2nd–order implementation, which uses five op-amps for each filter, there is a need of 1.5W. Given a 50% safety margin, a source capable of 2.25W should be used for such analog filters. In the case of the software processing, the Arduino and the sensor cumulatively consume approximately 400mW. The largest portion of the power is consumed by various unnecessary processes of the Arduino. If there was a need it could be lower with further optimization of the code used. We ought to take into consideration the fact that the

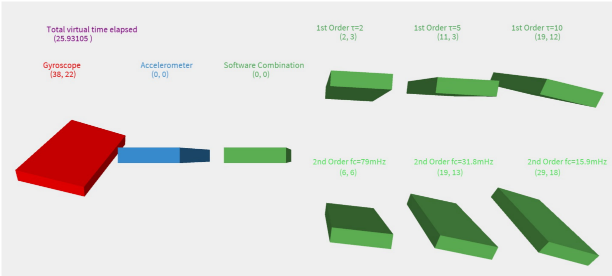


Fig. 7. Visual representation of gyroscope and accelerometer reported states, immediately after the sensor is relaxed after some movement, compared to the six differently tuned analog filters.

TABLE I
VALUES OF ROTATION IN THE CASE OF ZERO ROTATION OF THE SENSOR.

Rotation (°)	$t = 16.38s$		$t = 25.93s$		$t = 30.79s$	
	x	**y**	**x**	**y**	**x**	**y**
Gyroscope	13	11	38	22	61	29
Accelerometer	2	-11	0	0	0	0
Software combination	0	0	0	0	0	0
1^{st}–order ($\tau = 2s$)	0	3	2	3	6	2
1^{st}–order ($\tau = 5s$)	3	3	11	3	17	2
1^{st}–order ($\tau = 10s$)	7	8	19	12	29	13
2^{nd}–order ($\tau = 2s$)	2	-11	6	6	0	0
2^{nd}–order ($\tau = 5s$)	2	-11	19	13	0	0
2^{nd}–order ($\tau = 10s$)	2	-11	29	18	0	0

operational amplifiers need ±15V for their operation and use many cables and other components to function properly, in comparison to the Arduino board which only requires 3.3V or 5V but can also be used as the bare-bone IC design of the Atmel ATMEGA2560.

IV. CONCLUSIONS

The main derivation of this study is that for the filtering of accelerometer and gyroscope signals, the use of software processing is preferable. Only when the application requires the employment of analog design technology, then a 1^{st}–order complementary filter with τ=2s or lower can be considered. High attention should be put in selecting the correct cutoff frequency, for the chosen sensor, due to the different characteristics obtained for different values of the associated time-constants. Further analysis has to be done for other mainstream IMUs including those with a 3-axis magnetometer. With the latter one can obtain reliable yaw motion and be sure that, within reasonable timescales, the correct orientation could be maintained due to the earths magnetic field. Also, beyond the inside-out tracking done by the IMUs, GPS can be used for outside-in tracking, of any UAVs with line of sight to a GPS satellite. Further research steps include the employment of alternative techniques, such these presented in [16]–[20], for optimizing the performance of the system through the offered extra degrees of freedom in the filters frequency response.

ACKNOWLEDGMENT

The authors would like to thank C. Lagios for his help in the circuit construction, and I. Tsampras for his insightful thoughts in programming.

REFERENCES

[1] A. Pascoal, I. Kaminer, and P. Oliveira, "Navigation system design using time-varying complementary filters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 4, pp. 1099–1114, 2000.
[2] S. Colton and F. Mentor, "The balance filter," *Presentation, Massachusetts Institute of Technology*, 2007.
[3] M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel, "A complementary filter for attitude estimation of a fixed-wing UAV," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 340–345, IEEE, 2008.
[4] A. Jensen, C. Coopmans, and Y. Chen, "Basics and guidelines of complementary filters for small UAS navigation," in *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 500–507, IEEE, 2013.
[5] P. Marantos, Y. Koveos, and K. J. Kyriakopoulos, "UAV state estimation using adaptive complementary filters," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1214–1226, 2015.
[6] Q.-Q. Yang, L.-l. Sun, and L. Yang, "A fast adaptive-gain complementary filter algorithm for attitude estimation of an unmanned aerial vehicle," *The Journal of Navigation*, vol. 71, no. 6, pp. 1478–1491, 2018.
[7] M. G. Michailidis, M. Agha, M. J. Rutherford, and K. P. Valavanis, "A software in the loop (SIL) Kalman and complementary filter implementation on x-plane for UAVs," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1069–1076, IEEE, 2019.
[8] P. Narkhede, S. Poddar, R. Walambe, G. Ghinea, and K. Kotecha, "Cascaded complementary filter architecture for sensor fusion in attitude estimation," *Sensors*, vol. 21, no. 6, p. 1937, 2021.
[9] R. Cajo, T. T. Mac, D. Plaza, C. Copot, R. De Keyser, and C. Ionescu, "A survey on fractional order control techniques for unmanned aerial and ground vehicles," *IEEE Access*, vol. 7, pp. 66864–66878, 2019.
[10] G. Schmitz, T. Alves, R. Henriques, E. Freitas, and E. El'Youssef, "A simplified approach to motion estimation in a uav using two filters," *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 325–330, 2016.
[11] X. Wen, C. Liu, Z. Huang, S. Su, X. Guo, Z. Zuo, and H. Qu, "A first-order differential data processing method for accuracy improvement of complementary filtering in micro-uav attitude estimation," *Sensors*, vol. 19, no. 6, p. 1340, 2019.
[12] W. T. Higgins, "A comparison of complementary and Kalman filtering," *IEEE Transactions on Aerospace and Electronic Systems*, no. 3, pp. 321–325, 1975.
[13] A. Plummer, "Optimal complementary filters and their application in motion measurement," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 220, no. 6, pp. 489–507, 2006.
[14] T. Deliyannis, Y. Sun, and J. K. Fidler, *Continuous-time active filter design*. CRC press, 2019.
[15] M. Al-Shorman, M. Al-Kofahi, and O. Al-Kofahi, "A practical microwatt-meter for electrical energy measurement in programmable devices," *Measurement and Control*, vol. 51, p. 002029401879435, 08 2018.
[16] C. Coopmans, A. M. Jensen, and Y. Chen, "Fractional-order complementary filters for small unmanned aerial system navigation," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1, pp. 429–453, 2014.
[17] C. Coopmans, M. Podhradsky, and N. V. Hoffer, "Fractional-order complementary filters for small unmanned aerial system attitude estimation," in *2018 11th International Symposium on Mechatronics and its Applications (ISMA)*, pp. 1–7, IEEE, 2018.
[18] N. Sharma, E. Rufus, V. Karar, and S. Poddar, "Fractional order extended kalman filter for attitude estimation," in *International Conference on Intelligent Systems Design and Applications*, pp. 823–832, Springer, 2018.
[19] P. Bertsias, C. Psychalinos, and A. S. Elwakil, "Fractional-order complementary filters for sensor applications," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, IEEE, 2020.
[20] C. I. Muresan, I. R. Birs, E. H. Dulf, D. Copot, and L. Miclea, "A review of recent advances in fractional-order sensing and filtering techniques," *Sensors*, vol. 21, no. 17, p. 5920, 2021.