

Implementation in MATLAB of a Multiplicative Extended Kalman Filter for live estimation of a smart device's attitude

Javier Utrera* Rafael Vazquez**

* MSc Student in Aeronautical Engineering, Escuela Técnica Superior de Ingeniería, University of Seville, Seville 41092 Spain (e-mail: javutrcru@alum.us.es)

** Aerospace Engineering Department, Escuela Técnica Superior de Ingeniería, University of Seville, Seville 41092 Spain (e-mail: rvazquez1@us.es)

Abstract: This work introduces the tools used to teach the Kalman Filter (KF) to Aerospace Engineering students in the University of Seville. In particular, an easy-to-set-up application is introduced; based on the MATLAB framework on its 2020b (or newer versions), it is able to display the attitude of a smart device in real time through a wireless connection to a computer. This tool is a simple yet powerful educative resource when teaching about the KF, since it showcases its performance while allowing the student to understand how a complex real-world problem can be solved using a relatively simple implementation of the KF; in particular, the Multiplicative Extended KF (MEKF) is chosen, but the framework can be easily adapted to other versions such as the Extended or Unscented KF. In addition, the tool allows the student to be aware of the inner workings of the filter itself, learning about its advantages and limitations compared with other attitude estimation algorithms. The student is also able to understand how the filter needs to be tuned, and to observe the results of the experiments in a visual and straightforward manner.

Copyright © 2021 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: Extended Kalman Filters, Attitude algorithms, MATLAB, Education, Dynamics.

1. INTRODUCTION

The Kalman Filter (KF) is a powerful tool for optimal estimation of states using measurements coming from diverse sensors and corrupted by errors, and can be applied to many fields. In the field of Aerospace Engineering, it appears in many important applications, such as aircraft navigation, orbit determination, or attitude estimation. Thus, it is often included in Aerospace Engineering curricula, either on its own in a course on estimation theory, taught along control theory, or included as an advanced topic in a more applied class -for instance, in the University of Seville it is taught in the Spacecraft Dynamics course as an advanced tool for attitude estimation. Frequently, in the academic setting, the KF is taught and exemplified using fully determined models for the errors, thus resulting in completely defined problems -where the real state is also known. However, in real-world applications it is seldom the case that this error -and logically the real state- is fully known, and therefore the academic case may be deceitful (the student falsely believes that in practice they will just need to tweak some values to achieve a good result, but in reality oftentimes it is impossible to determine what a good result is, due to a lack of a reference), leaving the student defenseless when encountering a real problem for the first time. For this reason, it is desirable to provide the student with real examples, or to propose scenarios of

experiments that emulate real world conditions, so that they are forced to practice their critical thinking.

A prime example of a simple to understand and basic KF application -and therefore educationally very valuable- is the estimation of the attitude of a smart device (such as a smartphone or a tablet), which also has the benefit of being readily available for most students. The error of the sensors that are going to be used is not known a priori, and thus this error must be modelled in some way, similarly to a real-world case. The key advantage of this method is the ease of visualization of the results and the simplicity of performing as many experiments as necessary at any location, such as at the students' home. If done correctly, students will be able to improve their understanding about the filter's performance, by means of direct visual interpretation of the results, and about exactly how the tweaking of the different parameters of the filter is affecting the results.

The idea of using an smartphone for this purpose is not new. In Hendeby et al. (2017), a specific cell phone application was developed to carry out experiments. In this work, we follow the spirit of the latter work, but design a tailored application based on the MATLAB framework, and specifically oriented towards the 2020b or newer version, where support for live data transfer from a remote device has been reintroduced. The main advantage of using MATLAB is its wide availability in academic environments

and having both ends (smartphone end and computer end) with cross-platforms applications already developed and maintained by Mathworks; thus one can expect that the developed application can be used in teaching for many years to come. The Aerospace Engineering Department from the University of Seville already has good experiences in developing MATLAB-based educative applications, see e.g. Lanagran-Soler et al. (2015).

The structure of this work is as follows. A basic review of the Kalman Filter (KF) and the Extended Kalman Filter (EKF) is presented on the second Section. On the third Section, the problem of estimation of a smart device's attitude is proposed and the Multiplicative Extended Kalman Filter (MEKF) is presented. In Section 4, the solution to the problem and the different results that can be obtained using the designed application are shown. Finally, some concluding remarks are given in Section 5.

2. A REVIEW OF THE KALMAN FILTER AND SOME OF ITS NON-LINEAR VARIATIONS

The problem of state estimation on a system arises in real-world applications where measurements can be corrupted by error and the reliability of the sensors, as well as the presence of noise and bias and the imperfect modelling, needs to be taken into account. In its most general sense, the KF is a tool that provides estimations for the states based on a model and the available data at a certain time instant, combining measurements in such a way that the error of the estimation with respect to the real state is (hopefully) minimized.

The KF starts from a system's dynamic model. The system can have additional inputs as well. With information about the initial conditions of the system and the input function, it is possible to directly (by means of numerical integration) solve the system's model. However, the initial conditions and measurements will probably have some error, and the model will be imperfect. When numerically integrating a system using measurements and initial conditions that have uncertainty, these errors end up accumulating with time, potentially causing a drift from the correct state. After a certain time period, the uncertainty of the estimation of the state may become unacceptable.

An immediate solution for this problem is to introduce measurements from external sensors that are able to detect a magnitude related to the state, as a way of correcting the estimation. However, these measurements may be corrupted by error as well. The idea behind the KF is to combine the propagation of the system (integrating the system model) and the measurements from external sensors in such a way that the estimation is better than the estimate obtained by using only one measurement alone, or direct integration. As such, it is a common sensor fusion and data fusion algorithm.

Thus, one combines the estimation from the differential equation of the system's model (called the "a priori" estimation obtained from a "propagation step") with the available external measurement or measurements in an update step to obtain the best possible combination (called the "a posteriori" estimation). The combination is "best"

in the sense that it minimizes the covariance of the error (however this is not guaranteed at all in a nonlinear setting).

Next, the KF formulation is briefly explained. There are several possible choices to pose a KF depending on whether the dynamics of the system and the measurements are considered discrete or continuous. It is the authors' opinion that the continuous/discrete version, see e.g. Jazwinski (2007), is the easiest to teach from a pedagogical point of view, since students are more used to differential equations as models. However, what follows can be also adapted to a discrete/discrete version of the KF.

Starting with the linear case, consider thus the following description of a system:

$$\dot{x} = F(t)x + B(t)u + G(t)w, \quad (1)$$

where x is the state, u some input, assumed perfectly known (it could be the value of some internal measurement), and w models an unknown variable (which could come from error, perturbations, or unmodelled terms), assumed to be Gaussian white noise, this is: $E[w(t)] = 0$, $E[w(t)w^T(\tau)] = Q(t)\delta(t-\tau)$, where $E[\cdot]$ is the expectation and δ is Dirac's Delta function. In (1), the matrices F , B and G are of adequate dimension. The (discrete) measurement model is

$$y_k = H_k x_k + v_k, \quad (2)$$

where the subindex k refers to the discrete time instant t_k at which measurements are performed (this is, for $t \in (t_k, t_{k+1})$ no external measurement is performed). The value of y_k represents the value of the measurement at t_k , $x_k = x(t_k)$, H_k is a matrix of adequate dimension, and v_k is discrete Gaussian white noise, this is $E[v_k] = 0$, $E[v_k v_j^T] = R_k \delta_{kj}$. An additional hypothesis is that the noises in (1) and (2) are independent, this is, $E[v_k w(t)] = 0, \forall k, t$, and that an estimation of the initial condition of x at t_0 is known, denoted by \hat{x}_0 . The error of this initial condition is assumed Gaussian, of zero mean, independent of noises, and its initial covariance, $P_0 = E[(x(0) - \hat{x}_0)(x(0) - \hat{x}_0)^T]$ is assumed known.

The linear KF algorithm, which was shown optimal in R. Kalman's seminal paper (Kalman (1960)), is as follows:

- **Initialization:** Start from $\hat{x}_0^+ = \hat{x}(0)$ and $P_0^+ = P(0)$. Set $k = 0$.
- **Propagation:** For $t \in [t_k, t_{k+1})$, since no measurement is available, estimate $\hat{x}(t)$ by integrating (1) without the unknown noise: $\dot{\hat{x}}(t) = F(t)\hat{x} + B(t)u$ from the i.c. \hat{x}_k^+ . The covariance of the error $P(t)$, with the model being linear and all distributions assumed Gaussian, is known to follow

$$\dot{P}(t) = F(t)P(t) + P(t)F(t)^T + G(t)Q(t)G^T(t), \quad (3)$$

a matrix differential equation with i.c. P_k^+ .

- **Update:** When $t = t_k$ a measurement y_k is obtained. Denote the a priori estimation as $\hat{x}_k^- = \hat{x}(t_k^-)$ and the a priori covariance as $P_k^- = P(t_k^-)$. Then obtain the a posteriori estimation as

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - H_k \hat{x}_k^-), \quad (4)$$

where K_k , the Kalman gain, is obtained as

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (5)$$

The a posteriori covariance is updated as

$$P_k^+ = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \quad (6)$$

Set $k = k + 1$ and go back to the propagation step.

In the case of a nonlinear system, the following model is used:

$$\dot{x} = f(x, u, t) + G(t)w(t), \quad (7)$$

where now f is a function, and the measurement model becomes

$$y_k = h(x_k, t_k) + v_k, \quad (8)$$

with noises and initial conditions as in the linear case.

The easiest KF algorithm (but not necessarily the best performing one) is the Extended KF (EKF), see e.g. Welch et al. (1995), that uses the linear KF by linearizing the equations *around the current estimation* only for the purposes of computing $P(t)$ and the Kalman gain K_k . It is summarized next:

- **Initialization:** Start from $\hat{x}_0^+ = \hat{x}(0)$ and $P_0^+ = P(0)$. Set $k = 0$.
- **Propagation:** For $t \in [t_k, t_{k+1})$, estimate $\hat{x}(t)$ by integrating $\dot{\hat{x}}(t) = f(\hat{x}, u, t)$ from the i.c. \hat{x}_k^+ . Compute $F(t) = \left. \frac{\partial f(x, u, t)}{\partial x} \right|_{x=\hat{x}(t)}$ and using this F integrate (3) starting from P_k^+ .
- **Update:** At $t = t_k$, denote $\hat{x}_k^- = \hat{x}(t_k^-)$ and $P_k^- = P(t_k^-)$. Then:

$$\hat{x}_k^+ = \hat{x}_k^- + K_k \delta y_k, \quad (9)$$

where $\delta y_k = y_k - h(\hat{x}_k^-, t_k)$ and the Kalman gain is obtained again from (5) with $H_k = \left. \frac{\partial h(x, t_k)}{\partial x} \right|_{x=\hat{x}_k^-}$, and the a posteriori covariance from (6). Set $k = k + 1$ and go back to the propagation step.

3. ATTITUDE ESTIMATION

One application of the KF that is paramount in aerospace, easily understandable, and aids in visualizing its functionality and behavior, is the problem of attitude estimation through the combination of measurements of the angular velocity—from gyroscopes—and measurements provided by additional sensors such as accelerometers or magnetometers.

3.1 Attitude estimation model for a smart device

The attitude of a device with respect to a reference system needs to be estimated. For this, measurements of the angular velocities can be obtained from gyroscopes, whereas an accelerometer and a magnetometer can be used to obtain measurements for the direction of the gravity and of the magnetic field respectively. Note that for this scenario, it has been assumed that accelerations different from gravity are negligible, as well as perturbations to the magnetic field.

Then, it is possible to use any version of KF to integrate and combine, optimally, all measurements (or whichever are available at a given time instant). Specifically, since the problem is considered from the point of view of attitude and quaternions will be used as the state to represent the orientation, the Multiplicative EKF, see Markley (2003),

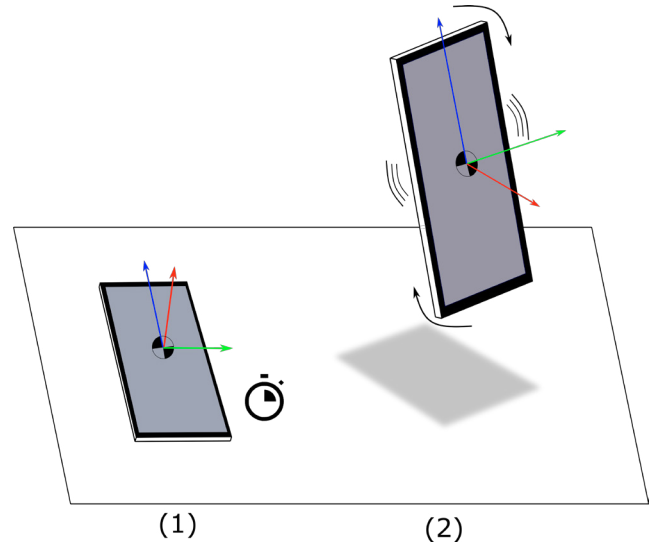


Fig. 1. Visualization of the experiment to be carried out. In the first step (1), measurements from the smartphone’s sensors at rest are taken, and the initial position—and reference system with respect to which the attitude will be expressed—are established. Then, the device is moved around (2), while the attitude estimation is being carried out to observe the results

which contemplates a multiplicative definition of the error, is used.

One needs to define the involved frames of reference, which will be denoted by A and B. Frame A refers to an inertial reference frame (as a first approximation, in fact, a local frame where the smart device is initially placed can be considered as “inertial” neglecting the extra inertial forces) and B to the body axes frame (see Fig.1). The attitude between the frames can be represented using several mathematical objects; of those, quaternions (q) and the director cosine matrix (denoted as C_A^B , representing the orthogonal transformation matrix from A to B) will be used in this work. Both frames are assumed initially aligned, thus the initial condition is the identity. If starting in a different orientation, the initial condition which can be obtained using any of the (static) attitude determination methods available such as TRIAD or Davenport’s Q method, see e.g. Shuster and Oh (1981). The state can then be propagated using the quaternion kinematic equation, and the update step can be performed using the measurements from the gravity \mathbf{g} and the Earth’s magnetic field \mathbf{b} using

$$\mathbf{g}^B = C_A^B \mathbf{g}^A \quad (10)$$

$$\mathbf{b}^B = C_A^B \mathbf{b}^A \quad (11)$$

First of all, the system would need to be calibrated so that the characteristics of each sensor (mean and covariance) can be estimated. The mean of the gyroscope will be taken as its bias, whereas the mean of the accelerometer and magnetometer measurements, as the reference \mathbf{g}^A and \mathbf{b}^A , respectively. Readings from the sensors would be obtained for a significant period of time so that the data obtained from the device at rest (which in the ideal case should yield that the device is indeed at rest) can be analyzed to extract the bias and the noise estimated from the measured data.

Of course it is possible to carry out the method without either the accelerometer or the magnetometer (if both sensors were left out, the procedure would no longer identify as a KF, but rather a direct kinematic integration), but the gravity measurement only yields high accuracy in determining the “down” direction and the magnetometer in determining the “North” direction, thus losing one sensor the ability to compensate the estimation around the three axis is lost.

MEKF equations for attitude estimation This section summarizes Markley (2003); a brief review of quaternions and the associated error vector is provided.

Quaternions extend the concept of complex number to 4 dimensions, in such a way that a quaternion can be written in the form $q = q_0 + iq_1 + jq_2 + kq_3$; where q_0 denotes the scalar part and $\mathbf{q} = [q_1 \ q_2 \ q_3]^T$ denotes the vector part. i , j and k are the basic quaternions, and the rules for multiplication amongst the basic quaternions can be defined as:

$$1i = i; 1j = j; 1k = k; \quad (12)$$

$$i * i = j * j = k * k = -1; \quad (13)$$

$$i * j = k; k * i = j; j * k = i; \quad (14)$$

$$j * i = -k; i * k = -j \ k * j = -i; \quad (15)$$

The most direct interpretation of quaternions as a means of attitude representation is as follows:

$$q_0 = \cos\left(\frac{\theta}{2}\right) \quad (16)$$

$$\mathbf{q} = \sin\left(\frac{\theta}{2}\right) \mathbf{e} \quad (17)$$

where \mathbf{e} and θ are Euler’s axis and angle respectively. If an original attitude is rotated said angle around the axis \mathbf{e} , one obtains the desired final attitude; thus, quaternions can be used to represent attitudes with respect to a determined reference system.

Quaternions are particularly well suited for numerical computation of attitude and kinematics, since the composition of rotations using quaternions is by means of direct component by component multiplication, eliminating the need for trigonometric functions, which are, in general, difficult to work regarding numerical computation of results. Also, even when numerical errors arise in the propagation of quaternion kinematics, it is possible to take advantage of one of the main property of quaternions - they are unitary-, so that through renormalization of the quaternions, numerical errors are contained.

Another property of quaternions is that the classical addition formulation used for description of errors and linearization cannot be used, since the sum of two unitary quaternions representing a reference and a slightly different attitude, respectively, does not necessarily have unitary modulus. Because of this, a multiplicative formulation is introduced, where $q = \bar{q} * \delta q$, with δq is the error quaternion that should be close to the unit quaternion $q = [1 \ 0 \ 0 \ 0]^T$. Note that since the error quaternion only

has 3 degrees of freedom, it can be codified using an error vector \mathbf{s} as:

$$\delta q(\mathbf{a}) = \frac{1}{\sqrt{4 + \|\mathbf{a}\|^2}} \begin{bmatrix} 2 \\ \mathbf{a} \end{bmatrix} \quad (18)$$

Which if \mathbf{a} is sufficiently small can be approximated as $\delta q(\mathbf{a}) \approx \begin{bmatrix} 1 \\ \mathbf{a}/2 \end{bmatrix}$. This small error vector can be shown to be equivalent to the traditional small error vector that is used to define the difference between two very similar DCM matrices:

$$C_A^B \approx Id - \theta^\times \quad (19)$$

if frames B and A are sufficiently close.

For the computation of the attitude of a body using the MEKF, the initial data required are $\hat{q}(0)$, initial attitude; $P(0)$, the initial covariance of the quaternion error vector \mathbf{a} ; Q , the covariance of the gyroscopes; and R_i , the covariance of each of the external measurements (in this case, the magnetometer and the accelerometers).

To begin with the estimation process, assuming that the measurements of the angular velocity of the device $\hat{\omega}$ are continuous, \hat{q} and P can be initialized for $t = 0$ using the initial conditions. Then, they are propagated until an external measurement is available, as

$$\dot{\hat{q}} = \frac{1}{2} \hat{q} * q_{\hat{\omega}} \quad (20)$$

$$\dot{P} = -\hat{\omega}^\times P + P \hat{\omega}^\times + Q \quad (21)$$

where $q_{\hat{\omega}}$ refers to a quaternion with zero scalar part and having $\hat{\omega}$ as vector part, and $*$ denotes quaternion multiplication. The operator \times transforms a vector into the matrix that implements the vector product, such that $a^\times b = a \times b$, see e.g. Markley (2003).

Then, at a time instant $t = t_k$ when a measurement is received, the a priori estimation is computed as $\hat{q}^- = \hat{q}(t_k)$ and $P^- = P(t_k)$. Next, it is possible to compute the discrepancy δz_i between the external measurement and the a priori estimation from the reference value v_i^A through

$$\delta z_i = \hat{v}_i^B - \hat{C}_A^B(\hat{q}^-) v_i^A \quad (22)$$

Where the index i refers to each of the n external sensors (in this case, 2) and \hat{C}_N^B is the estimated Direction Cosine Matrix (DCM) for the current estimated attitude obtained through the Euler-Rodrigues formula for quaternions

$$C(q) = (q_0^2 - \mathbf{q}^\top \mathbf{q}) Id + 2\mathbf{q}\mathbf{q}^\top - 2q_0 \mathbf{q}^\times \quad (23)$$

After computing δz_i , $H_i \approx (\hat{v}_i^B)^\times$ and R_i , these are joined by blocks into the vector δz and matrices H and R , and the Kalman gain and a posteriori quaternion error vector \mathbf{a}^+ can be calculated as

$$K = P^- H^\top (H P^- H^\top + R)^{-1} \quad (24)$$

$$\mathbf{a}^+ = K \delta z \quad (25)$$

Finally, it is possible to update the estimation for the attitude and its covariance error using

$$\hat{q}^+ = \hat{q}^- * \delta q = \hat{q}^- * \begin{bmatrix} 2 \\ a^+ \end{bmatrix} \frac{1}{\sqrt{4 + \|a^+\|^2}} \quad (26)$$

$$P^+ = P^- - KHP^- \quad (27)$$

and then this point can be used as the initial condition for the next propagation until an additional measurement arrives. Note the algorithm is flexible enough to accommodate for the presence of 1, 2 or more measurements, becoming available at different times, individually or together.

4. IMPLEMENTATION OF THE ATTITUDE ESTIMATION APPLICATION

In order to provide the student with a simple to work with yet powerful and easy to understand tool, a MATLAB based application with a graphical user interface has been developed. This application allows the student to tweak the parameters of the MEKF while being able to gain insight about how the problem is being solved. For the application, the class `OrientationView` from the app described in Hendeby et al. (2017).

The application consists of a window with options that automate the connection to a mobile device through the MATLAB mobile app on a mobile device. For this, an active Mathworks account is required, as well as MATLAB r2020b (or newer) for live transmission of data from the sensors. An active internet connection on both ends is also necessary, but said devices can be on separate networks.

It is possible to access the workspace variables at anytime during the exercise, and the MATLAB application code is also provided, so that the students can tweak the application even more, having a say on how the calibration step is carried out, and if different models for the sensor errors are to be considered, like a more complex approach to determine their bias.

The workflow of the exercise would be to:

- (1) Open up the MATLAB Application and the MATLAB Mobile App
- (2) Set up the experiment by placing the mobile device on a flat surface that will serve as an initial reference position and defines the reference frame A.
- (3) Connect to the mobile device
- (4) Once the connection is confirmed, select the duration for the calibration step
- (5) Perform the calibration step and observe the results, with the device at rest conditions and away from magnetic interference which may distort readings from the magnetic sensor
- (6) Establish the values for the matrices Q and R, which roughly correspond to the weight given to the measurements of the different sensors on the computation of the attitude
- (7) Determine a duration for the attitude estimation experiment
- (8) Perform the experiment (click start)
- (9) Observe the results as the device is gently rotated
- (10) Set the phone in the initial reference position before the experiment ends
- (11) Take note of the calculated error quaternion once the experiment is over, and the estimated error, which ideally should coincide

In the proposed application, it is possible to compare the direct integration method with the MEKF, as well as with the smart device's own orientation readings.

In an academic setting, the values of Q and R should be tweaked by the student to minimize the error, and to reduce the discrepancy between the real and estimated errors as much as possible. Note that in the application, one unique value of R has been specified, but in the code it is possible to determine a different weight for the magnetometer and for the accelerometer. It is interesting to study how different sensors found in different mobile devices affect the quality of the results, and if it is possible to mitigate this effect by taking it into account when pondering each sensor differently in the MEKF. It is also possible to remove measurements from one of the discrete sensors, as to observe the effect on the results.

A lot of variables can affect the quality of the results of the experiments. The duration and quality of the calibration can be crucial in modelling correctly the characteristics of the sensors that are being used, which also affects the values of the Q and R matrices. The nature and duration of the experiment itself, along with the selected refresh rate, also have a significant contribution to the way the algorithm behaves. If a low refresh rate or fast movements are exerted on the mobile device, it is possible that the hypothesis of continuous measurements of the angular velocity is no longer valid. If significant accelerations are exerted on the device, the readings of the accelerometer will be unreliable.

The developed application runs in real-time, but requires significant computing power, so if need be it is possible to reduce the rate at which the representation is updated, so that the attitude's computation is not slowed down and compromised due to the non-critical visualization phase.

The application is available under request to the authors.

5. CONCLUSION

A MATLAB GUI-based interactive application has been developed, which will allow students to gain insight on how quaternion kinematics and attitude estimation equations can be implemented in an experimental environment, while also providing an in-depth understanding of how a Kalman Filter works, what its variants are and how to apply it to more complex problems than what may be possible to teach by traditional means.

The final objective of this application is integration inside a dedicated lab session in the context of Kalman Filters and attitude estimation. To design said activity, it would be interesting to have the student recreate experiments with variations in the different application parameters and to have them identify what factors are relevant, what configuration yields the best results for a set of experiments and finally to determine the limitations of such algorithm, all while interactively tampering with the inner workings of the open-source application.

Future works also include implementing other KF variants (e.g. UKF) to compare between them and see the advantages and drawbacks of each, or developing the application in other frameworks such as Python.

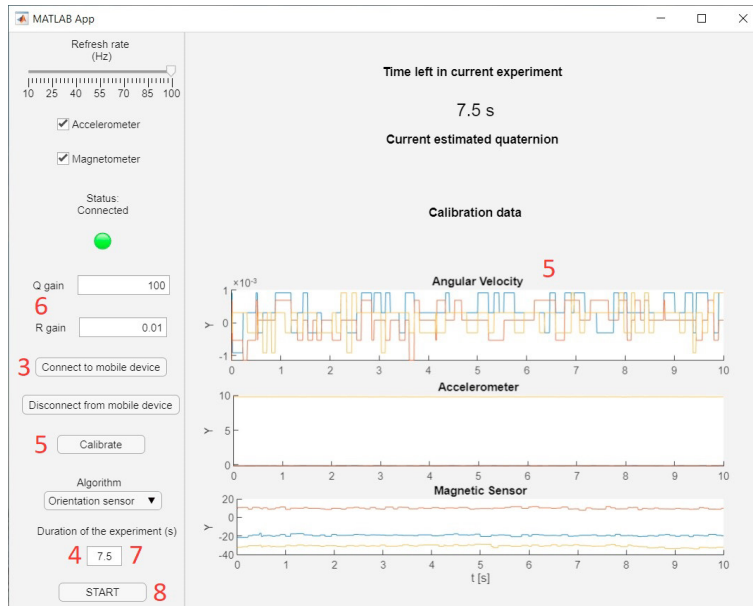


Fig. 2. General view of the tailored MATLAB application and of the results of the calibration phase, including the different steps necessary to set up and carry out the experiments

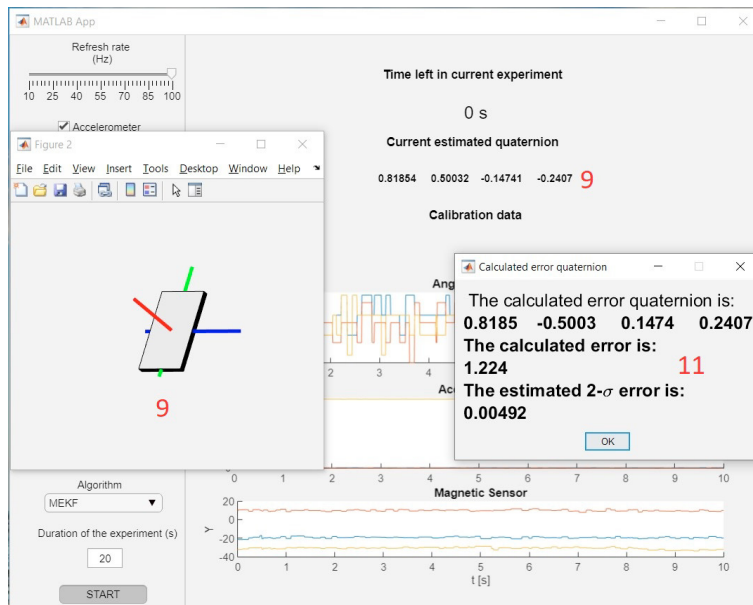


Fig. 3. Visualization of the device's attitude and final errors

ACKNOWLEDGEMENTS

Rafael Vazquez acknowledges the funding from Spanish Ministerio de Ciencia, Innovación y Universidades (grant PGC2018-100680-B-C21).

REFERENCES

- Hendeby, G., Gustafsson, F., Wahlström, N., and Gunnarsson, S. (2017). Platform for teaching sensor fusion using a smartphone. *International journal of engineering education*, 33(2B), 781–789.
- Jazwinski, A.H. (2007). *Stochastic processes and filtering theory*. Courier Corporation.
- Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D), 35–45.
- Lanagran-Soler, F., Vazquez, R., and Arahal, M.R. (2015). A Matlab educational GUI for analysis of GNSS coverage and precision. *IFAC-PapersOnLine*, 48(29), 93–98.
- Markley, F.L. (2003). Attitude error representations for Kalman filtering. *Journal of guidance, control, and dynamics*, 26(2), 311–317.
- Shuster, M.D. and Oh, S.D. (1981). Three-axis attitude determination from vector observations. *Journal of guidance and Control*, 4(1), 70–77.
- Welch, G., Bishop, G., et al. (1995). An introduction to the Kalman filter.