SENSOR PARAMETER ESTIMATION FOR THE PERFORMANCE ENHANCEMENT OF AN AEROSPACE APPLICATION

Jairo Eduardo Moraes Siqueira (M.Sc.)
MECTRON E.I.C. S.A. (ODT – Odebrecht Defense & Technology S.A.), jairo.siqueira@gmail.com1

Hélio Koiti Kuga (Dr.)
INPE/DMC (National Institute for Space Research/Division of Space Mechanics and Control), hkk@dem.inpe.br2

Marcelo Lopes de Oliveira e Souza (Ph.D.)
INPE/DMC (National Institute for Space Research/Division of Space Mechanics and Control), marcelo@dem.inpe.br

Abstract: *Current systems such as: unmanned aerial vehicles-UAVs, satellites, aircrafts, automobiles, turbines, power controls and traffic controls are becoming increasingly complex and/or highly integrated as prescribed by the SAE-ARP-4754 Standard. Such systems and their control systems use many modes of operation and many forms of handling information to achieve high levels of performance and high levels of accuracy under changing environments and phases of their lifecycle. So, the adoption of solutions that favour lower costs, either for development or operation, considering both the hardware and the software in use is mandatory. Therefore, to improve the economic benefit for a certain application, the accuracy that certain classes of sensors might lack can be overcome by properly performed calibration. In this paper, the Discrete-Discrete Kalman Filter (DDKF) is used, firstly, for the calibration of a baroaltimeter of a UAV with the help of other more accurate sensors (e.g. GPS, IMU, etc.). Secondly, the calibration performed is verified, in the same fashion. The baroaltimeter is embedded in a semi-autonomous aerospace vehicle, which flight is comprised of two phases: captive (for the calibration task) and free (for the verification task). To accomplish the proposed goals, one must take into account: 1. the adopted filtering technique; 2. the boundary operational conditions and initial values for the system' mission; 3. the estimation of sensor parameters which affects the accuracy such as the random bias and the scale factor for the sensor in question. The formulation and equationing for the filter, and the sensor model are described, implemented computationally, and then tested for such a set of representative situations. The results obtained highlight: 1. the difficulties for tuning the KF parameters and conversions; 2. the KF accuracy in estimating true values of unknown parameters; 3. the usefulness of this technique in enhancing the performance of systems quick and accurately*..

Keywords: *Kalman Filter, Fault Detection, Flight Control.*

1 Introduction

The use of UAV's (Unmanned aerial vehicles) has become widespread (i.e., beyond military and defense applications), not only in scale, but in diversity of fields as well: borders surveillance, maintenance of high voltage transmission grids, fauna and flora integrity control in game reserves, quick reaction anti-crime police units, and so forth [RECASCINO, 2006].

This is, yet partially, due to key features UAV's present, like:

- Flexibility (no need of prepared runways);
- Ease-of-use (no need of expensive ground crews for operation/maintenance) and
- Cost attractiveness (cheaper than propellers or helicopters).

However, the increasing demand is not a one-fold issue, which derived itself from "naturally" growing portfolios consisting of present existing and future potential users. There is an "underneath" demand, directly related with the type of technical solutions adopted, since these definitely impact on economic issues related to the vehicles success. By its turn, this success could be stated as two-fold:

- Development: by fulfilling the design specifications and overcoming the design restrictions within the budgeted time and costs;
- Production/procurement and operation/maintenance: the type of solution adopted above will produce a share of the recurrent costs and the vehicle's pricing (in production/procurement), and of the sustaining costs (in operation/maintenance).

UAV's, for instance, do represent a class of complex systems, for having other complex subsystems integrated within and with its airframe (e.g. its navigation-guidance-piloting, radio-frequency command, power supply, external sensors, and propulsion), all of them having some sort of embedded software.

Taking into account the considerations above, the use of less sophisticated (hardware) components combined with adequate (software) calibration strategies might allow a realization capable of meeting technical-operational requirements established for the overall system, even under limited budgetary conditions (financially or time-wise).

In this paper, the Discrete-Discrete Kalman Filter (DDKF) is used, firstly, for the calibration of a baroaltimeter of a UAV with the help of other more accurate sensors (e.g. GPS, IMU, etc.). Secondly, the calibration performed is verified, in the same fashion.

To do that, this work is structured as follows:

- Section 2 formally states the problem and its scope;
- Section 3 briefly exposes the Kalman Filter concept, which is adjusted for implementing the object of study. Then part of the achieved results are shown and discussed;
- Section 4 performs a deeper analysis, focusing on limitations found and possible solutions for further research;
- Section 5 synthetizes all previous sections according to the proposal brought by the Abstract

2 Problem Definition

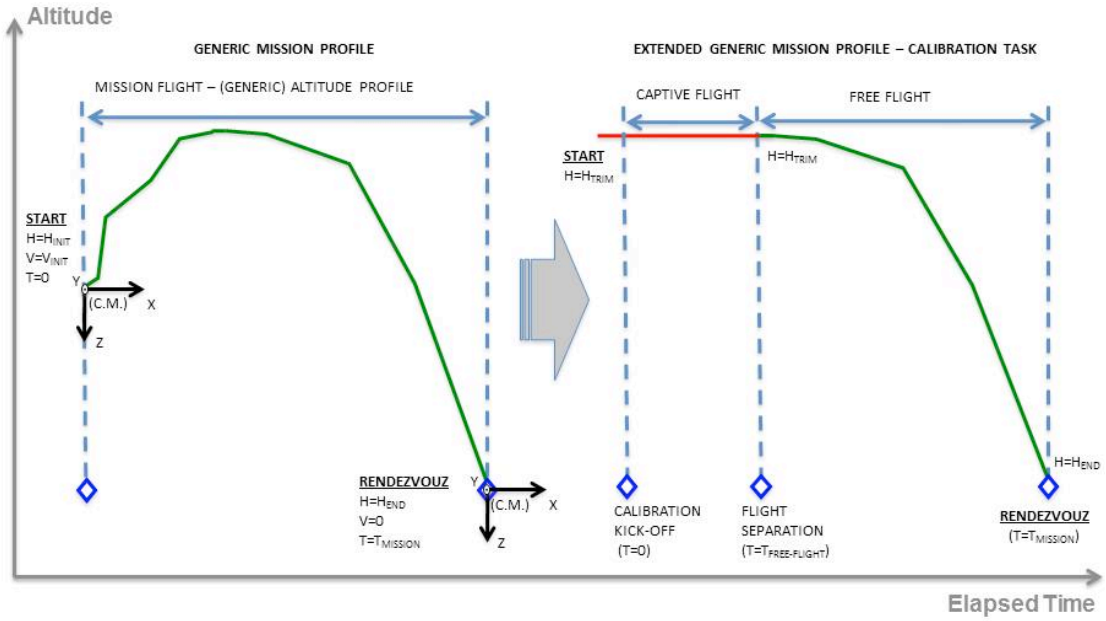
Consider a generic, mission-oriented UAV (which could be a propeller, a rocket, a glider, and so forth) having a mission profile similar to that depicted by Figure 1 (left to the arrow), below. From the starting point of its mission (air or ground launched), it must locate and track a specific signal emitted at the rendezvous point (which could be a laser, or a radio-frequency one, and so forth), and right afterwards, it must move towards it.

In this work, navigation itself is enabled by two different sensors used by the UAV: one IMU (Inertial Measurement Unit), for the guidance and piloting control loops, and one baroaltimeter, which tracks the altitude exclusively. Although, and theoretically, the IMU itself would be enough for the UAV's altitude tracking, the use of a baroaltimeter helps to constrain position error growth in the local vertical channel, during navigation [WENDEL ET ALLI, 2006].

However, the baroaltimeter itself has its own inaccuracies that must be taken into account, so the observations performed can be properly used for the navigation purposes. In other words, the baroaltimeter must be calibrated. Systematic error components commonly found in literature are the scale factor and bias [BIEZAD, 1999 AND SAVAGE, 1996]. In this work, only the bias is considered, and this as a time-invariant parameter to be estimated. Furthermore, its susceptibility to dynamics effects (i.e., velocity and acceleration) will be investigated.

To gather invaluable data for comparison and analysis, the calibration process is performed for the UAV IMU as well.

Therefore, on the right hand side of Figure 1, the calibration methodology for the UAV sensors is roughly depicted.



. **Figure 1. Generic mission profile for problem definition**

Attached to an aircraft (working as a "mother-ship") by an under-wing pylon, and equipped with both a GPS (Global Positioning System) receiver and a telemetric unit, the UAV is put to fly trimmed (constant altitude and airspeed). After a given elapsed time (necessary for the calibration to occur), the UAV is launched for a simulated mission. One can clearly note two phases of flight, captive and free, during which data are constantly generated and transmitted via telemetry.

These data must be thereafter analyzed to reproduce (or at least, try to) the calibration executed during the captive flight, as well as the UAV performance (or the lack thereof) during the free flight. From now on, this work shows what was implemented for the on-line (real) process, and for the verification (simulated) process as well.

3 Calibration via the Discrete-Discrete Kalman Filter

3.1 The Discrete-Discrete Kalman Filter

The Kalman Filter estimates a process by means analogous to feedback control: the filter estimates the process state and observations at a certain time step, then right afterwards it obtains the "feedback" in the form of the difference between the process (noisy) observations and the filter estimates of them[MAYBECK, 1979 AND GELB, 1974]. The Kalman filter equations are split into two types: time propagation equations and time update ones. The time propagation equations project forward (in time) the current state and state error covariance estimates, so *a priori* estimates for the next time step are obtained. The time update equations are responsible for the "feedback", incorporating a new observation into the *a priori* state and state error covariance estimates, so an improved *a posteriori* state and state error covariance estimates are obtained.

Therefore, for a given observable dynamic system, its state-variable and output (continuous domain) equations are given by [MAYBECK, 1979]:

$$\dot{x}(t) = A(t)\cdot x(t) + B(t)\cdot u(t) + \Gamma(t)\cdot w(t); \tag{1}$$

where

$\dot{x}(t)$: [n x 1] States derivatives vector;
$A(t)$: [n x n] Transition matrix;
$x(t)$: [n x 1] States vector;
$B(t)$: [n x m] Input transmission matrix;
$u(t)$: [m x n] Input vector;
$\Gamma(t)$: [n x s] Noise-addition matrix;
$w(t)$: [s x 1]System uncertainty, white-noise vector, $w = N(0, Q)$.

$$y(t) = H(t)\cdot x(t) + I\cdot v(t); \tag{2}$$

where

$y(t)$: [p x 1] Observation vector;
$H(t)$: [p x n] State-observation relationship matrix;
$x(t)$: [n x 1] States vector;
I: [n x n] Identity matrix;
$v(t)$: [p x 1] Observation uncertainty, white-noise vector, $v = N(0, R)$.

Additionally:

Initial state: $x_0 = N(\bar{x}_0, P_0)$;
t: $\forall t \in [t_0, t_f], t_0 \geq 0$.
$w(t), v(t)$: Independent, uncorrelated noises. **(3)**

Now considering that the system is linear time-invariant, the equivalent equations for the discrete time-domain can be written, separating the deterministic and the stochastic variables, as [MAYBECK, 1979]:

[Propagation phase]

$$\overline{x}_k = A\hat{x}_{k-1} + Bu_{k-1}, \text{ for } \forall k \in \mathrm{N}, k \geq 0, \quad \textbf{(4)}$$

where

$\overline{x}_k$: [n x 1] Propagated state vector, k-th instant;
A: [n x n] Transition matrix;
$\hat{x}_{k-1}$: [n x 1] Updated state vector, (k-1)-th instant;
B: [n x m] Input matrix;
u_{k-1}: [m x n] Input vector, (k-1)-th instant;

$$\overline{P}_k = A\hat{P}_{k-1}A^T + G_k Q_k G_k^T \quad \textbf{(5)}$$

where

$\overline{P}_k$: [n x n] Propagated covariance matrix, k-th instant;
A, A^T: [n x n] Transition matrix and its transpose;
$\hat{P}_{k+1}$: [n x n] Updated covariance matrix for the (k-1)-th instant;
G_k, G_k^T: [n x s, s x n] Noise addition matrix and its transpose, k-th instant;
Q_k: [s x s] White sequence covariance matrix, k-th instant.

[Update phase]

$$K_k = \overline{P}_k H_k^T (H_k \overline{P}_k H_k^T + R_k)^{-1}, \quad \textbf{(6)}$$

where

K_k: [n x n] Updated Kalman gain-matrix, k-th instant;
$\overline{P}_k$: [n x n] Propagated state covariance matrix, k-th instant;
H_k, H_k^T: [n x n] Observation-state relationship matrix, k-th instant, and its transpose;
R_k: [n x n] Observation covariance matrix, k-th instant;

$$\hat{x}_k = \overline{x}_k + \left(y_k - H_k \overline{x}_k\right), \quad \textbf{(7)}$$

where

$\hat{x}_k$: [n x1] Updated state vector, k-th instant;
$\overline{x}_k$: [n x1] Propagated state vector, k-th instant;
y_k: [p x 1] Observation vector, k-th instant;
H_k: [p x n] Observation-state relationship matrix, k-th instant;

$$\hat{P}_k = \left(I - K_k H_k\right)\overline{P}_k, \quad \textbf{(8)}$$

where

$\hat{P}_k$: [n x n] Updated state covariance matrix, k-th instant;
I: [n x n] Identity matrix;
K_k: [n x n] Updated Kalman gain-matrix, k-th instant;
H_k: [p x n] Observation-state relationship matrix, k-nth instant;
$\overline{P}_k$: [n x n] Propagated covariance matrix, k-nth instant.

3.2 Problem Statement

The state, and the observation vectors are defined as follows: *noise*

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} h_{GPS} \\ bias_{IMU(Master)} \\ bias_{IMU(Slave)} \\ bias_{BARO} \end{bmatrix} = \begin{bmatrix} h_{true} + bias_{GPS} + noise_{GPS} \\ h_{IMU(Master)} - h_{GPS} \\ h_{IMU(Slave)} - h_{GPS} \\ h_{BARO} - h_{GPS} \end{bmatrix} \tag{9}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} h_{GPS} \\ h_{IMU(Master)} \\ h_{IMU(Slave)} \\ h_{BARO} \end{bmatrix} = \begin{bmatrix} h_{true} + bias_{GPS} + noise_{GPS} \\ h_{true} + bias_{IMU(Master)} + noise_{IMU(Master)} \\ h_{true} + bias_{IMU(Slave)} + noise_{IMU(Slave)} \\ h_{true} + bias_{BARO} + noise_{BARO} \end{bmatrix} \tag{10}$$

Since, as stated in section 1, the calibration process is performed with the "mother ship" trimmed (*quasi* constant altitude and airspeed), simplifying hypotheses can be stated for the set of equations (4) to (8):

- The GPS is considered, at this stage, to be unbiased;
- The process is linear, invariant. Therefore the transition matrix has the identity form;
- The process does not have any control inputs;
- The process dynamics should be disturbed by white-process in the observation;
- The observation-related uncertainty above is time-invariant as well;
- The relationship among states and observation (matrix H) is invariant;
- The observation and states from/for the "mother ship" are included in the overall calibration system, since the UAV is attached to it during the calibration;
- The initial values (3) are chosen arbitrarily (a priori) for the initial propagation.

Therefore, the matrices related to the problem become:

$$x_0 = \begin{bmatrix} 2100 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \quad ; P_0 = \begin{bmatrix} 50^2 & 0 & 0 & 0 \\ 0 & 100^2 & 0 & 0 \\ 0 & 0 & 150^2 & 0 \\ 0 & 0 & 0 & 300^2 \end{bmatrix} [\text{m}^2]; \quad R = \begin{bmatrix} 25^2 & 0 & 0 & 0 \\ 0 & 50^2 & 0 & 0 \\ 0 & 0 & 70^2 & 0 \\ 0 & 0 & 0 & 150^2 \end{bmatrix} [\text{m}^2]$$

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}; \quad \begin{aligned} G &= [0] \\ Q &= [0] \\ B &= [0] \end{aligned}.$$

And the set of equations (4-8) can assume the form of (11-15) below:

$$\bar{x}_k = A\hat{x}_{k-1} \tag{11}$$

$$\bar{P}_k = A\hat{P}_{k-1}A^T \tag{12}$$

$$K_k = \bar{P}_k H^T (H\,\bar{P}_k H^T + R\,)^{-1} \tag{13}$$

$$\hat{x}_k = \bar{x}_k + \left(y_k - H\,\bar{x}_k\right) \tag{14}$$

$$\hat{P}_k = \left(I - K_k H\,\right)\bar{P}_k \tag{15}$$

The initial values for the state vector can be defined *a priori*, for the calibration process to start, and are set according to the "mother ship" trimming conditions and the UAV operational requirements:

The related values for 'P' (initial) and 'R' for the UAV's sensors (IMU and baroaltimeter), the telemetry's GPS, and the "mother ship" IMU are obtained from their manufacturers' datasheets:

Equations (11-15) and the initial conditions above were implemented for the UAV, whose calibration (captive) flight lasted about 70 seconds. The sampling rate was of 200 Hz, and the first 10 (ten) seconds were neglected in order to avoid residual spiking that occasionally could emerge from the mother-ship's trimming.

For the results herein presented, there were used MATLAB/Simulink scripts and telemetry data from an actual flight, as explained in section 2. The results are depicted in Figure 2 and Figure 3[1].

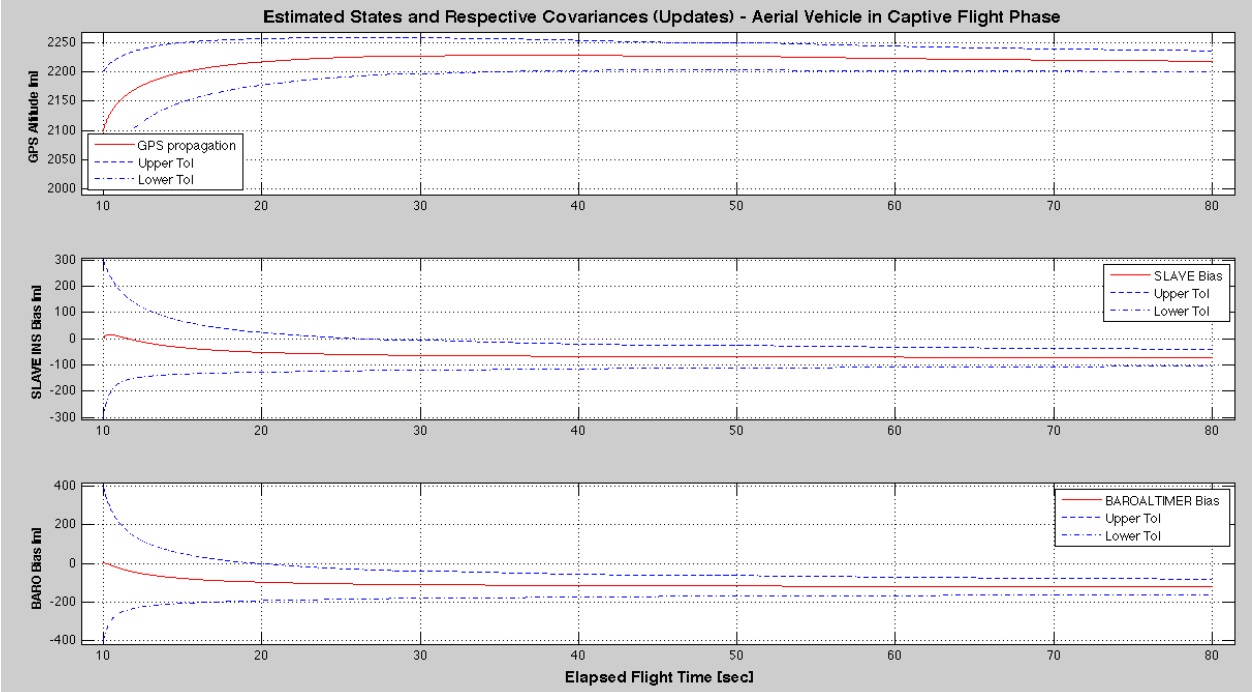


Figure 2. Results for the estimated parameters of the UAV sensors[2]

The calibration process quickly converged for all defined states (Figure 2), even for the baroaltimeter bias, which is clearly noisy. Although the convergence lasted up to the last time step of the calibration (captive) flight, it is really noticeable for the first 10 seconds (from t=10 to t=20 seconds).

Since it is a requirement for the calibration that the "mother ship" stays trimmed at 'constant' altitude, the speed of convergence means that the algorithm worked well for the proposed task, and that the chosen/used values for 'P' and 'R' were realistic.

The final values for the biases and respective covariances (obtained from the last time step during captive flight) for the "slave" IMU and the baroaltimeter present in the UAV are synthetized at Table 1 below.

Table 1. Experimental results for the "slave" IMU and baroaltimeter altitude observation biases, and related covariances.

SENSOR	Bias [m]	Covariance [m^2]
"Slave" IMU	-72.8909	$(\pm 32.4422)^2$
Baroaltimeter	-121.8691	$(\pm 40.3308)^2$

Figure 3 (right below) clearly confirms the information given by Figure 2. The values depicted are obtained from the relationship existing in equation (9),which are known as the 'residuals' of the estimation performed.

These 'residuals' are merely the difference between the sensors calibrated observations and those from the reference GPS. The residuals depicted for the GPS can be explained by the associated to the uncertainty of its observations. Again, the level of noise present in the baroaltimeter observation can be clearly noticed.

[1] From the flight test, the first 10 (ten) seconds are scrapped off to avoid excessively noisy observations, which may be unrepresentative for the calibration process proposed.

[2] In the figure, the reader may notice that instead of IMU (Inertial Measurement Unit), the acronym INS (Inertial Navigation System) is used. Although this is an inaccurate assumption (e.g., an IMU might be a component within an INS), for the limited ends of this work, one can consider both as synonyms.

As elapsed time reaches t=80 seconds, the UAV acquires free flight, which has a pattern of its own. Because of that, the sensors will be subjected to effects not experimented during calibration (e.g. velocity and acceleration ones). The following step is assessing for how long the calibrated observations will remain reliable.

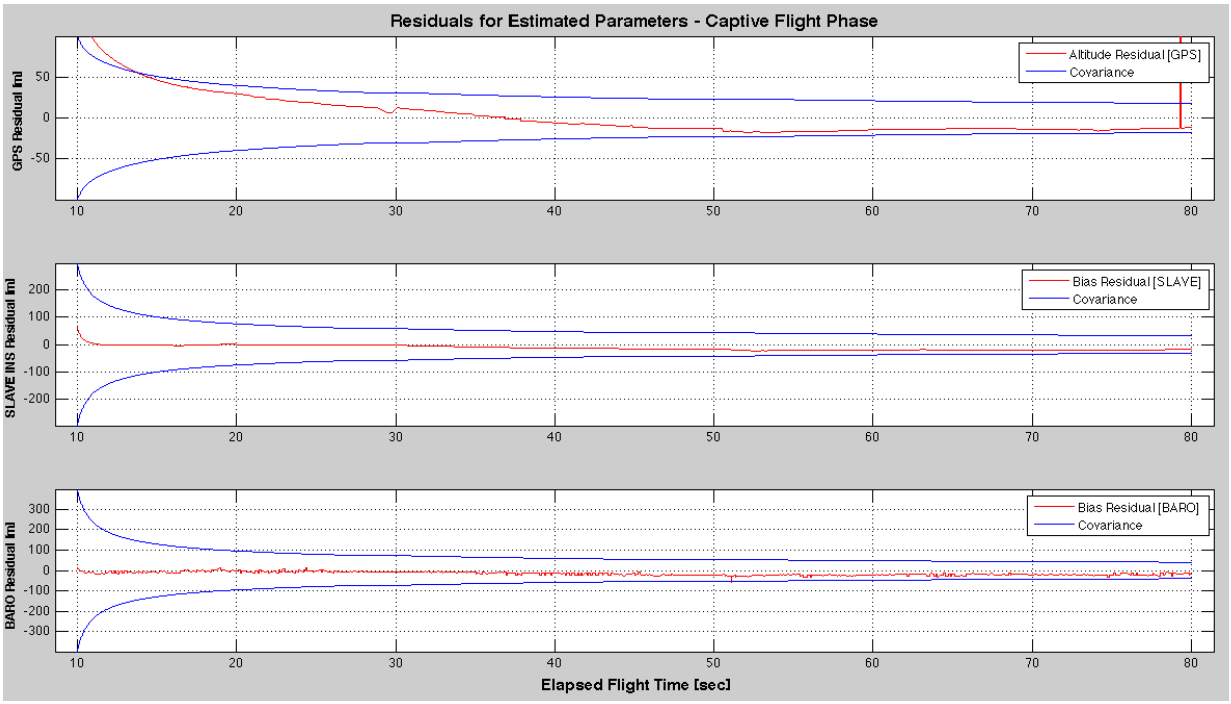


Figure 3. Filtered (estimated) observations residuals.

4 Verification during Free-Flight

This section verifies how well the sensors perform after being calibrated and are in free flight, which happens right after 80 seconds of captive flight. Thus, the following expression is evaluated for both the "slave" IMU and the baroaltimeter:

$$Sensor\_Error_k^i = y_k^i - bias^i - y_k^{GPS}; \tag{16}$$

where:

$SensorError_k^i$: Error of the i-th sensor observation, k-th instant;

y_k^i: i-th sensor gross observation, k-th instant;

$bias^i$: i-th sensor bias;

y_k^{GPS}: GPS gross observation, k-th instant;

Expression (16) must be subjected to the requirement:

" For the sensor observation to be reliable under free flight, there shall be always satisfied the condition

$$-\sigma^i \le Sensor\_Error_k^i \le +\sigma^i \;\; AND \;\; -\sigma^i \le MovingAverage(Sensor\_Error_k^i) \le +\sigma^i; \tag{17}$$

[3] "

If the left hand-side condition from (16) is, at any moment, once broken, the sensor observation becomes doubtful. Afterwards and at any moment, if the right hand-side condition is broken, the observations have become completely untrustworthy.

The reason for having chosen the moving average is simply because it acts as like a low-pass filter, thus not taking into account high-frequency inputs (or "spikes"). These "spikes" might warp the numerical interpretation of the physical phenomena of altitude variation during the free-flight. Furthermore, the algorithm for its implementation is relatively cheap.

[3] Where σ^i is the deviation due to each sensor, as shown by Table 1.

Additionally, having two different sensors allows the use of the criteria above to determine whether the information provided is still reliable. Figure 4, as follows, depicts each sensor performance during free flight.

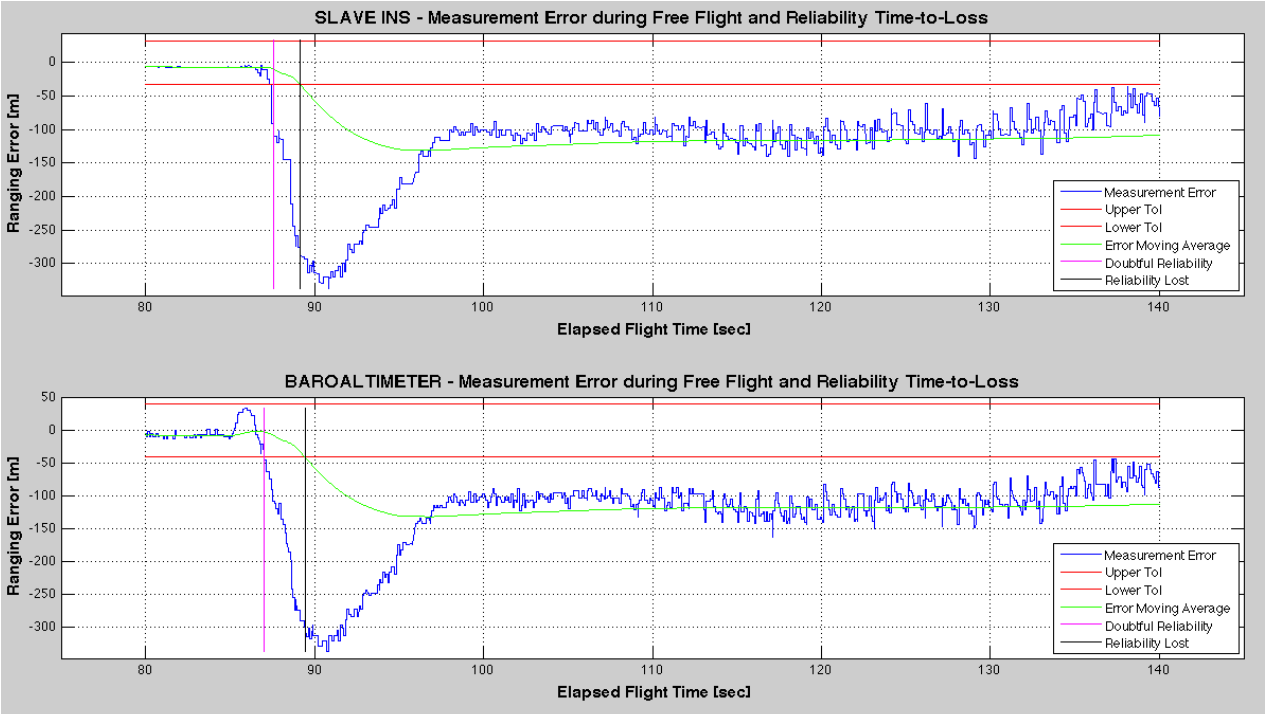


Figure 4. Error in the sensor observations during free flight.

Examining the results for the "slave" IMU, one can notice that (as shown by the magenta vertical line) the observation becomes doubtful after 7.6 (seven dot six) seconds of free-flight. The reason is because the first criterion of (16) is broken: the sensor error crosses the boundary established by the calibration process during captive flight. Slightly after 1.5 (one dot five) seconder later, it definitely drifts away, and breaks the second criterion of (17), related to the moving average (green curve, crossed by the black vertical line). One easily notices also that as the UAV keeps flying, the moving average seems to converge -100 meters.

The same analysis can be applied for the baroaltimeter, however, with poorer results. Roughly after 7 (seven) seconds of free flight, the observations become already doubtful (magenta vertical line), and 2.5 (two dot five) seconds later, breaks the second criterion of (17) (black vertical line). The same as for the "slave" IMU, the moving average (green curve) of the observation error seems to converge to -100 meters.

Considering both sensors, hardly one can say the results were good. The *quasi* static calibration process, which did not take into account dynamic effects (velocity, acceleration) into account, produces results with poor performance in conditions that differ from that of the calibration itself.

A quick-fix that could be tried out is inserting a second correction given by the moving average, by feeding it back into the observations performed by both sensors. If the UAV is supposed to operate under well-behaved and slow dynamics, this quick-fix could be incorporated into a pre-flight calibration tool. Conditions for engagement of this additional correction and variations thereof must also be studied and tested, as well as the affordability of such approach faced the mission requirements and profile. The use of a Kalman Filter should be also considered here, instead of using a moving average.

As an alternative to a second correction, it should also be investigated that one should rather perform the previous (and single) calibration with a complete model of the UAV's flight and sensors dynamics, and considering data (sensors) fusion.

5 Concluding Remarks

The problem of improving the performance of sensors used in UAVs, with the aim of making them more accurate and efficient with the help of software techniques was shown, yet for a limited scope. Further on, the technique (DDKF algorithm) to be used for such a task and the correlated assumptions were presented as well.

The set of sensors to be calibrated was chosen and had the algorithm applied, by simulating the performed flight test with the telemetry data available. Out of that, it was possible to assess how adequate the chosen method was for general performance of the then calibrated sensors.

Requirements for free flight observation error, actions for possible correction thereof, and alternatives for further investigation were proposed.

It became clear that the approach must be revisited and have its sophistication improved, faced the limited performance shown. The DD Kalman Filter has proved its capability, having shown to work properly even under strict limitations.

Acknowledgements

The authors would like to thank the Division and Course of Space Mechanics and Control (DMC/CMC) of the National Institute for Space Research (INPE) for the supplied infrastructure in hardware and software resources.

References

Biezad, D., 1999, Integrated Navigation and Guidance Systems, AIAA Education Series.
Gelb, A., 1974, Applied Optimal Control, The MIT Press, Cambridge, MA.
Maybeck, P. S., 1974, Stochastic Models, Estimation and Control, Volume 1, Academic Press.
Recascino, C. F., 2006, "The First Word: New Uses for UAV's", Strata: Research at Embry-Riddle Aeronautical University. 15 Jan 2012, <http://www.erau.edu/research/strata_pdf/strata_spring06.pdf>.
Savage P., 1996, Introduction to Strapdown Inertial Navigation Systems, Strapdown Associates.
Wendel, J.; Meister, O.; Schlaile, C.; Trommer, G. F., 2006, "Ein integriertes GPS/MEMS-IMU Navigationssystem für einen autonomen Helikopter", Aerospace Science and Technology (Elsevier), nr. 10, pp. 527–533.