Sensor Fusion with Cointegration Analysis for IMU in a Simulated Fixed-Wing UAV

Elias de Souza Gonçalves* and Paulo Fernando Ferreira Rosa†
*Instituto Militar de Engenharia, Rio de Janeiro, Brasil, e-mail: *si.elias.ti@gmail.com*
†Instituto Militar de Engenharia, Rio de Janeiro, Brasil, e-mail: *rpaulo@ime.eb.br*

***Abstract*— This work deals with the problem of navigation and location of an unmanned aerial vehicle (UAV) that results in the estimation of the state variables of the vehicle. In order to solve the problem, first we chose a fixed-wing aircraft model available for flight simulators. Then, we applied the cointegration method to a set of inertial sensors composed of two accelerometers, and two gyroscopes. It is an analytical technique to verify common trends in multivariate series and long-term and short-term dynamic modeling. That allows us to discover the best readings from one IMU, or if not possible, we can find out the best features of each IMU working together. Thus, the inherent contribution of this work is the use of cointegration as a way of estimating the behavior of the UAV inertial sensors. In the last step, a widely used tool for the prediction of state variables, the Extended Kalman Filter (EKF), merges the sensors of the previous step with the GPS to eliminate inaccuracies. A software in the loop architecture is proposed as a validation methodology. The result shows that the estimates of the state variables were satisfactory, always remaining close to those considered true and calculated by the embedded software.**

***Keywords*— cointegration; state estimation; filtering; multisensor fusion**

I. Introduction

A considerable amount of unmanned aerial vehicle applications has been used for different purposes in civilian and military areas [1], [2], [3]. All these applications, in order to perform autonomous flight adopt hardware that has various sensors that can measure environment signals and can estimate the actual vehicle state. The state of a UAV is composed of: (a) motion parameters, such as position, velocity and acceleration; and, (b) attitude parameters, such as heading angle, tilt, rotation angle [4]. Because of the particularities of each sensor, some have a low update rate of dynamic navigation. When this occurs the sensor may not be available at all times, thus compromising navigation.

In order to overcome this problem, a vast number of articles has been published about sensor fusion for aerial and terrestrial robots. Most of them addressing problems like simultaneous localization and mapping (SLAM) to get a better state estimate in an unknown environment [5], [6], [7], either by formulating the problem as a factor graph or even by using the least squares formulation to minimize the error. Some implementations use additional sensors or multi-sensors for fault tolerance [8], [9]. In general, the Kalman Filter (KF) is widely used since Rudolf Kalman introduced it in 1960 [10]. Its special variations like Extended Kalman Filter (EKF) was discussed in [11], [12], [13], and Unscented Kalman Filter (UKF) was addressed in [14] and [15].

Differently from the above mentioned works, we explore the use of cointegration and the error vector correction model. For the sensors of a UAV, it should be considered a non-linear approach. However, the two inertial sensors of the UAV have homogeneous characteristics, so we adopt the linear model used in [9] with an Extended Kalman Filter.

In summary, in this paper we present an approach in which low cost sensors only are used to perform cointegration test in two inertial measurement units (IMU) in order to get the best estimation for inertial parameters. This applies to acceleration and rotation measurements; and, later on, the EKF performs sensor fusion. In this step the diagonal process noise measurement matrix R is the standard deviations from accelerometer and gyroscope in 3d-axis given by cointegration test.

Moreover, with this statistical tool we can discover the relation between pairs of sensor signals from the architecture. That allows us to discover the best readings from one IMU, or if not possible, we can find out the best features of each IMU working together. Also, one signal can be substituted by others when there is that relation, culminating in a more assertive vehicle state estimation.

Furthermore, we validate our approach with a simulated flight test. Finally, some results are presented, followed by our conclusions and future works.

II. The Deployement Of a UAV

In this section, we define the necessary requirements to assemble robust and multipurpose UAV.

A. Aircraft Configuration: A Paraglider

For this work, we use a simulated model of the aircraft in Figure 1. That model is radio-controlled with 4-channel. Also, it is manufactured with CNC cuts and finished with colored adhesive tapes. It has fiberglass stringers in the extension of the wing for reinforcement purposes. The wing is removable to access the electronics compartment and it is attached to the fuselage with elastic rubber. The airplane requires basic assembly, like attaching the stabilizers and wing support.

B. Mainboard And Autopilot

As the autopilot board, we use a Raspberry Pi based autopilot Navio2, from Emlid (Figure 2a), a hardware

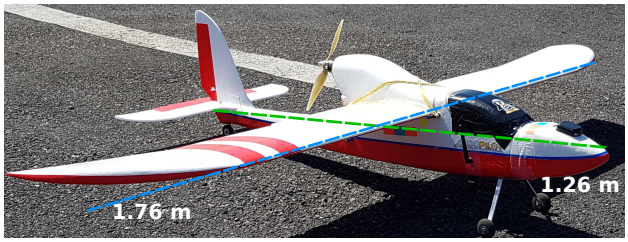
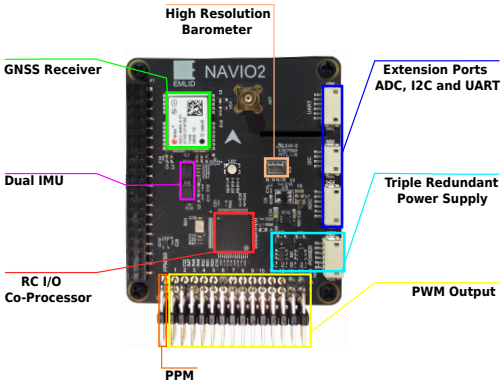


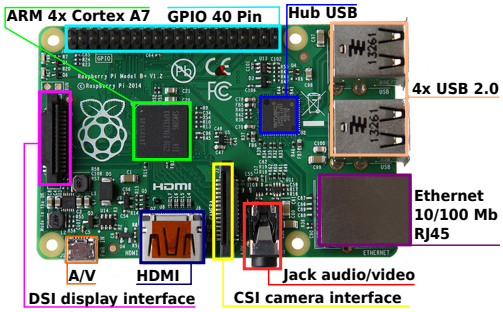
Figure 1. Fixed Wing Model.

chosen to comply with the proposed design requirements. The autopilot is designed to work with the microcomputer Raspberry Pi (Figure 2b) which is capable of real-time processing.

The Navio2 is equipped with sensors able to perform autonomous flight test, with 2 IMUs, that provides three axis accelerometer, three axis gyroscope, and three axis magnetometer each. Also has barometer, GPS, PWM, PPM, serial communication UART, I2C, SPI, and ADC. It is a compact board with dimensions of 55 mm x 65 mm, and weight 23 g.



(a) Navio2 autopilot board.



(b) Microcomputer Raspberry Pi.

Figure 2. Mainboard and autopilot.

Furthermore, the microcomputer required to use Navio2, Raspberry Pi, in its version 2 Model B features a robust and suitable hardware which has a 900 MHz 32-bit ARM Cortex A7 processor, 1GB of RAM, 4 USB ports, 40 GPIO pins, camera interface (CSI), micro SD card slot and videoCore IV 3D graphics core. In our work, we use a Linux kernel with real time capabilities, and as the autopilot base software, the ArduPlane [16], which is part of the free Autopilot Source Project.

C. Architecture

Our architecture comprises the FlightGear software simulator and JSBSim, respectively as flight view and UAV dynamic controller for simulation. Besides that, the APM Planner 2 as the ground control station of the mission, and the API Dronekit as the development tool, which has encapsulated the protocol MAVLink, that allows communication among all software.

III. The Problem Statement

The localization problem of a UAV can be formulated using an IMU to measure the UAV acceleration and rotation rates. At the inertial navigation system these measurements are processed and transformed by navigation equations which have been considered in the NED (North, East, and Down) local frame [17] to provide the position, velocity and attitude of the vehicle. In this work, we had been using two IMUs to improve the quality of navigation. As shown in the flowchart of the proposed inertial navigation system (Figure 3), we have been executed cointegration test in acceleration and rotation measures. The cointegration block receives measures from two IMUs (in red IMU1 and in blue IMU2) and perform a unit root test. If the measurements are stationary, it runs the regression; otherwise, it runs Johansen's test cointegration using an error correction model. It also provides velocity, position, and attitude moving from the body frame to the navigate frame. Despite the dual aspect, we want to capture the best readings from one IMU or, if not possible, we need the best features of each IMU working together. Thus, two navigation systems are introduced at the next subsection: INS, that is developed for a wide range of vehicles, and GPS, that nowadays is the most used.

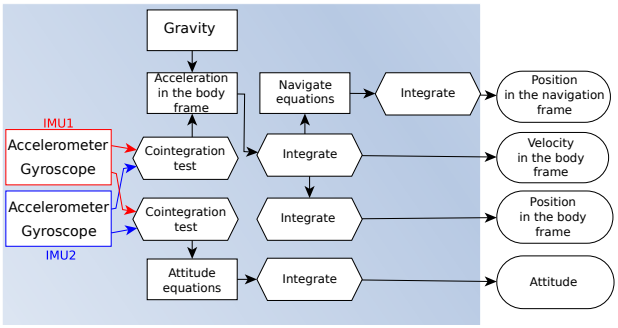


Figure 3. Proposed inertial navigation system.

A. GPS / INS Integration

In general, these systems have complementary characteristics present in two scenarios. First, the development of an inertial system requires reliable and effective extra information to minimize errors. Second, the GPS itself presents errors, the most commons are due to clock offset of the receiver, signal propagation through ionosphere and troposphere, beyond receiver noise errors and multipath.

For scenario one, the accelerations and angular rates provided by an IMU are integrated with the INS algorithm and that integration accumulates several errors such as accelerometer bias, gyros drift, temperature, and vibrations present in the INS system. GPS provides additional information and highly accurate position and velocity using measurements range. In the second case, INS is extremely useful because it gives conditions to navigate in denied GPS environments. Then, the best estimates of the aircraft states are possible by GPS / INS integration as shown Figure 4, where the input of EKF is the prior state vector given by inertial navigation system and the GPS measures which take the position of the UAV. The EKF corrects the inertial state estimate with GPS position and takes the state estimate post correction.

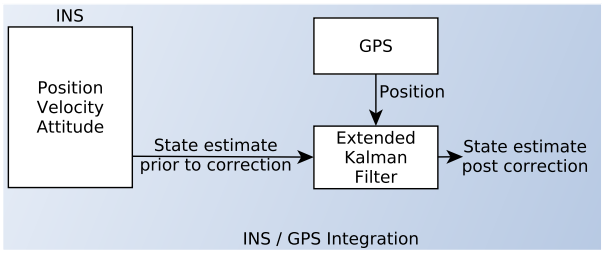


Figure 4. Block diagram INS / GPS integration.

Equation 1 gives the state vector of the vehicle where (X, Y, Z) are the positions, (U, V, W) are the velocities and (ϕ, θ, ψ) are the Euler angles. Equation 2 represents the output vector where (p, q, r) are the angle rates and (ax, ay, az) are the accelerations [18].

$$x = [X, Y, Z, U, V, W, \phi, \theta, \psi]^T. \tag{1}$$

$$u = [p, q, r, ax, ay, az]^T. \tag{2}$$

In order to navigate, we must define two reference frames, the body or inertial reference frame which represents the vehicle and the navigation frame, which as the name suggests, is for the vehicle motion. Then, the system equations of motion can be given by simple integrations and frame transformations. Firstly, by Euler equations in [18] we get Euler rates $(\dot{\phi}, \dot{\theta}, \dot{\psi})$.

In this work we have used two IMUs at the vehicle center of gravity to measure acceleration and rotation rates. However, we have considered just one vector (u) after performing cointegration test. Equation 3 gives $(\dot{U}, \dot{V}, \dot{W})$, which are the true vehicle accelerations in the body frame.

$$\begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} = \begin{bmatrix} ax + Vr - Wq + g\ sin(\theta) \\ ay - Ur + Wp - g\ cos(\theta) sin(\phi) \\ az + Uq - Vp - g\ cos(\theta) cos(\phi) \end{bmatrix}. \tag{3}$$

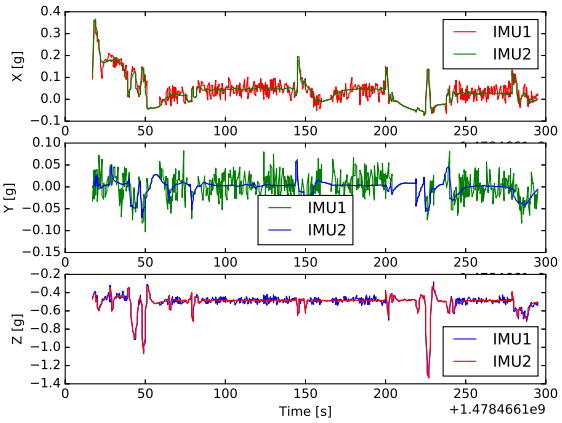
In order to obtain the velocity of the vehicle (U, V, W) in the body frame, we need to integrate the acceleration vector. When we integrate the velocity vector we get the position of the vehicle in the body frame. Then, the velocity needs to be transformed by the coordinate transformation matrix and to be integrated to obtain the position vector $([X, Y, Z]^T)$ in the navigation frame, as in Equation 4:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \int C^T bn(\phi, \theta, \psi) \begin{bmatrix} U \\ V \\ W \end{bmatrix} dt. \tag{4}$$

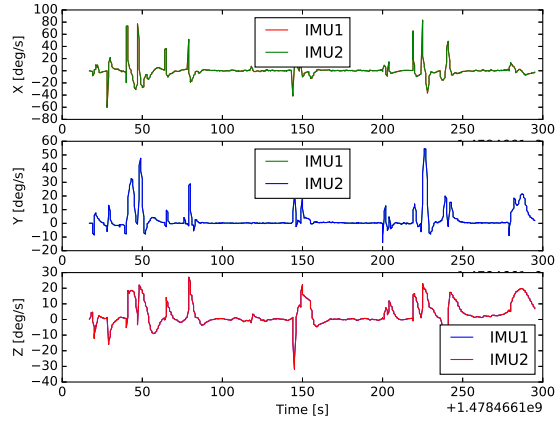
The direction cosine matrix (C_{bn}) rotates a vector from the body frame to the navigation frame [18].

B. Cointegration

In this paper, we introduced cointegration on a sensor signal from two IMUs in a proposed inertial navigation system. Then we build a multi-sensor architecture with two accelerometers and two gyroscopes. Figure 5 shows the IMUs readings for the usual sensors for this architecture. We use time series analysis in a range of data and a statistical method [9] where one signal can be substituted by others when there is a relation between sensor signal pairs from the architecture. Cointegration is the statistical tool used for discovering this relationship.



(a) Accelerometer.



(b) Gyroscope.

Figure 5. Measurements from two-sensor group IMU in 3 axis.

Not only stationary signals are present in a sensor group output, but also non-stationary noise. In order to keep

the trend of the signal data, we do not use difference calculation to achieve stationarity. Like [9] points out, "is very difficult to identify the pattern of non-stationary signals", then we cannot examine the signals directly, but we can find the relation among all signals and use it in our model by Equation 5:

$$y_{1,t} = \alpha + \beta y_{2,t} + \varepsilon, \tag{5}$$

where α and β are the regression parameters, $y_{1,t}$ and $y_{2,t}$ are the sensors and the subscripts refer to the IMU number and time. Finally, ε is the residuals of regression.

Augmented Dickey-Fuller test (ADF) [19] is the unit root test adopted for analyzing the stationarity of series and the Johansen test to estimate the cointegration relation. In order to be cointegrated, each series needs to be individually integrated, and their relation needs a lower order of integration. Equation 6 presents the unit root test employed:

$$\Delta y_t = \beta_1 + \beta_2 t + \delta y_{t-1} + \sum_{i=1}^{m} \alpha_i \Delta y_{t-i} + v_t, \tag{6}$$

where β_1 is the intercept series, β_2 is a trend coefficient, δ is the unit root coefficient, m is the lag order of the autoregressive process, α_i is a parameter of the model and v_t is the white noise. We test the null hypothesis given by $H_0 : \delta = 0$ and perform regression of Δy_t in $y_{t-1}, \Delta y_{t-1}, \ldots, \Delta y_{t+p-1}$ to calculate t-statistic in each time series. The critical values of t-statistic are calculated by [19].

Johansen [20] proposes an approach considering two likelihood ratio tests: the trace test and the maximum eigenvalue test. We chose the trace test, that analyze the number of independent linear combinations of time series variables by Equation 7:

$$J_{trace} = -T \sum_{i=r+1}^{n} ln(1 - \hat{\lambda}_i), \tag{7}$$

where T is the number of observations, ln is the log of the estimated eigenvalues, $\hat{\lambda}$ is a canonical correlation, r is the number of cointegrating vectors, and n is the number of alternative hypothesis of cointegrating vectors.

IV. Cointegrated Angle Rates And Acceleration

The measurements of low-cost MEMS sensors are sensitive and noisy. Thus, obtaining the resulting acceleration of the force applied to the UAV from a single inertial sensor may not be sufficient.

This approach considers measurements of two IMUs, both MPU-9250 (IMU1) and LSM9DS1 (IMU2) operating at a rate of 100 Hz with 16 bits data output. The navigation data is previously stored during the flight test. We need to extract the reading of the navigation data and the attitude from accelerometers and gyroscope on the Navio2 controller board.

A. Calibration

This is a method which removes structural errors in the sensors output, improving sensor performance and reliability of the measurements from IMUs. Thus, the accelerometers were calibrated by Equation 8, keeping the UAV stall at six different positions: left side, right side, level, nose down, nose up, and back side. In this way, the Gravity is the total force applied on the UAV body. Table I shows the biases used in each accelerometer axis to improve performance and accuracy of the sensors.

$$\beta = acc_{measured} - sensitivity * acc_{actual}, \tag{8}$$

where $\text{acc}_{measured}$ is the measured accelerometer data, sensitivity is the factory calibrated sensitivity scale that is assumed to be 16,384 LSB/g (least significant bit per g) according with typical specifications in datasheets, and acc_{actual} is the value of accelerometer in each above-mentioned position (which is affected by the sensitivity of the accelerometer along the actual axis and is assumed to be 1 g).

TABLE I
ACCELEROMETERS BIASES.

Bias axis	MPU-9250	LSM9DS1
β_x	-0.0042	0.0099
β_y	0.0291	-0.0020
β_z	-0.0226	-0.0068

B. Filtering And Fusion

We have been created a Python program with three modules. Module one was implemented and embedded in UAV to intercept and to store MAVLink messages in real time navigation. Module two, shown in Algorithm 1, is a cointegration test applied to the inertial measurements resulting from Module one. Lastly, Module three is an EKF implementation.

The Algorithm 1 works like a pre-filter. The procedure starts by applying the correction factor bias in both measurements. After the data are organized in matrix form, then the process is created. The time series (y_t, x_t) are tested concerning unit root by ADF test. If the test is false x_t and y_t are both stationary and the algorithm estimates them by regression. If the test is true, y_t and x_t are both non-stationary, then there exists a linear combination of them. Now the Johansen (object $jres$) is applied for the process with autoregressive order equals 1, also the cointegration vectors are given and applied in y_t and x_t for correction.

Finally, the standard deviation (σ^{imu}) of the error is calculated and returned together with the estimated time series ($\hat{xy}_t$). That output values are used for sensor fusion with EKF where the GPS measurements are integrated with INS measurements as previously shown in Figure 4. Then, $\hat{xy}_t$ is the cointegrated measurements from IMUs, and σ^{imu} composes the main diagonal of process noise matrix R. Section V shows the results after application of

cointegration test and EKF in measurements previously stored during a real flight test.

Algorithm 1: Cointegration test for two time series

Input : y_t: time series from first IMU, x_t: time series from second IMU, τ: time in microseconds, β: correction factor

Output: $\hat{xy_t}$: the estimated time series from input, σ^{imu}: standard deviation of residual error

```
y_t, x_t ←applyFactor(y_t, x_t, β)
data ← (y_t, x_t)
proc ← (τ, data)
If testADF(y_t) and testADF(x_t) then
    jres ←testJohansen(proc, 1)
    v ←jres .evecr
    y_tcorrect, x_tcorrect ←jres .np.dot(proc, v)
    ε, x̂y_t ←Regression(y_tcorrect, x_tcorrect)
end-if
Else
    ε, x̂y_t ←Regression(y_t, x_t)
end-if
σ^imu ←getStandardDeviation(ε)
Return σ^imu, x̂y_t
```

V. Experiments

We conducted a simulated flight test. The objective was to perform an automatic flight: At this stage, it would be verified the viability of an autonomous flight with the autopilot software ArduPlane in a simulated environment. An algorithm for storing MAVLink messages with the data from the flight was executed simultaneously.

After this step, the Algorithm 1 was executed three times for acceleration and three times for angle rates, which considers a pair of accelerometers and a pair of gyroscopes in each axis. The result of ADF test is shown in table II, where the line elements organize all sensors, lines two and three display the p-values of sensors respectively from IMU1 and IMU2. From line five ahead, we have the critical values. When the p-value is bigger than the critical value, we have a scenario of non-stationary series. This occurs with the sensor acc_z from IMU1 and IMU2. For all other sensors, we have a stationary scenario.

TABLE II
ADF test results of sensors from IMU1 and IMU2

sensors	acc_x	acc_y	acc_z	gyro_x	gyro_y	gyro_z
p-values	-6.75	-4.72	-0.95	-12.27	-6.39	-5.37
	-7.21	-4.72	-0.86	-12.27	-6.39	-5.37
critical-values						
1%	-2.56					
5%	-1.94					
10%	-1.61					

We applied the Johansen trace test for acc_z sensors from IMU1 and IMU2 with the critical values of 95%. Then we find that the values in the second row (trace test) of the table III are bigger than the values in the fourth row (critical values at 95%) of the same table. As shown by ADF test, Johansen confirms that the series of the process are non-stationary. However, Johansen identifies the cointegration vectors that allows the series to become stationary.

TABLE III
Johansen test results of sensors acc_z from IMU1 and IMU2

sensor	acc_z	
trace test	489.29443629	152.81086446
critical-values		
95%	15.4943	3.8415
cointegration vectors		
v1	65.68323835	-66.60901815
v2	-9.4182437	-1.46024865

The estimated time series and the residual errors after execution is shown in Figure 6, where the time series of IMU1 are represented in blue and the time series of IMU2 are in green. In the magenta color, we have the estimated time series given by the model, and in red the residual error. It is verified that the estimated time series is always close to the IMU1 measurements. That is because we use the time series of the first IMU as dependent variables; in other words: that is what we really want to predict. An exception is shown in Figure 6c, where the estimated time series is the mean value of two original series after reaching the stationarity.

The standard deviations of the error presented in table IV completes the process noise covariance matrix R that models the uncertainty of the measurements.

TABLE IV
Standard deviation from residual error

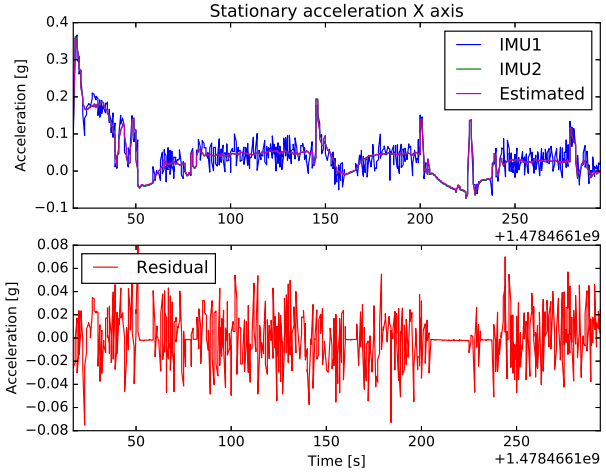
	x	y	z
σ^{imu}_{acc}	0.02153	0.02220	0.84824
σ^{imu}_{gyro}	0.01864	0.01849	0.01929

Figure 7 shows the states of the vehicle after applying the EKF in measures from the flight test with our approach compared with the sates given by ArduPlane autopilot.

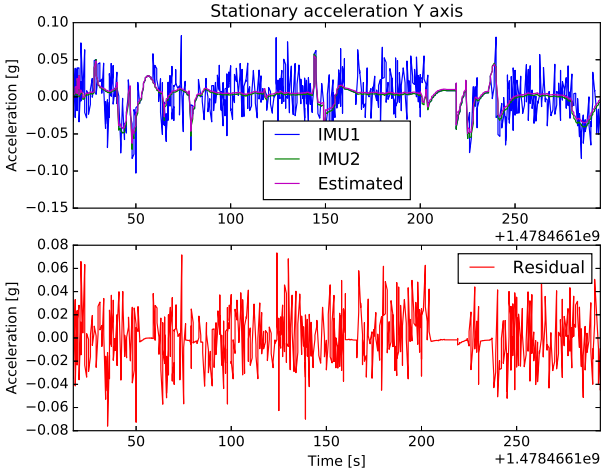
VI. Conclusion

In this study, a statistical method was adopted to analyze measurements from inertial sensors of a UAV as a time series. Also, an EKF was employed with less emphasis to improve the estimation of UAV states, that has as observations all the analyzed inertial measurements and measurements from GPS.

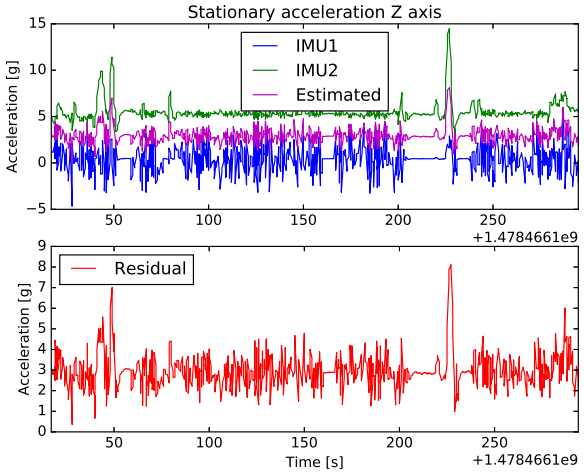
The results have shown that our approach is capable to find the relation of cointegration in each pair of inertial sensor from this UAV. The algorithm ensures the stationarity of the time series by applying ADF and Johansen tests. Thus, by regression, we estimated the relation of cointegrated measurements, giving us the accurate state estimation after applying EKF.



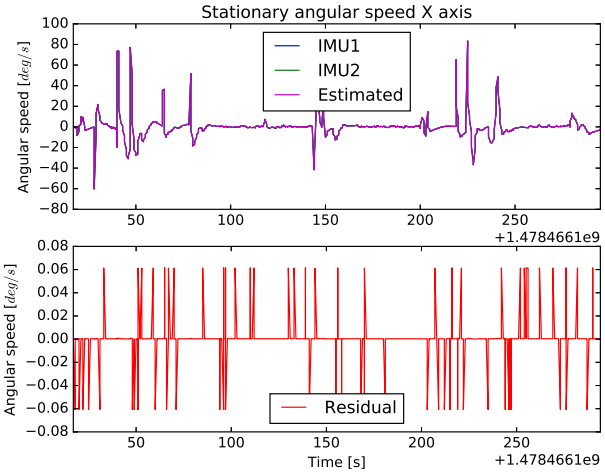
(a) Accelerometer x.



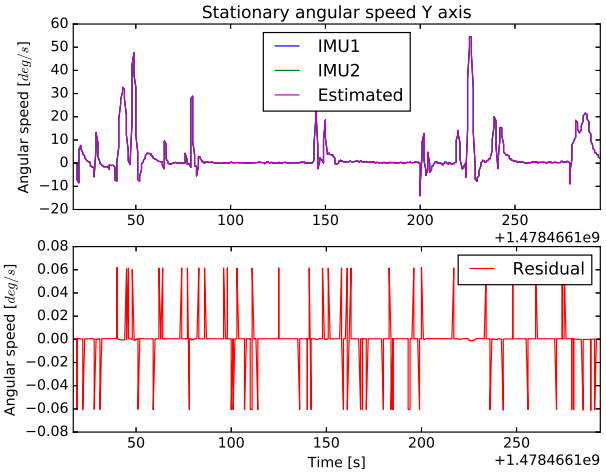
(b) Accelerometer y.



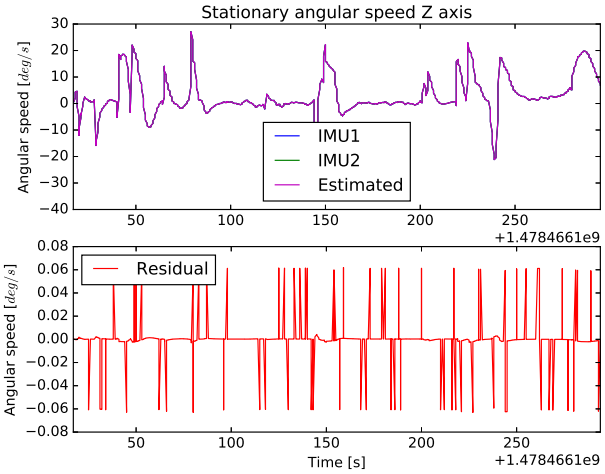
(c) Accelerometer z.



(d) Gyroscope x.



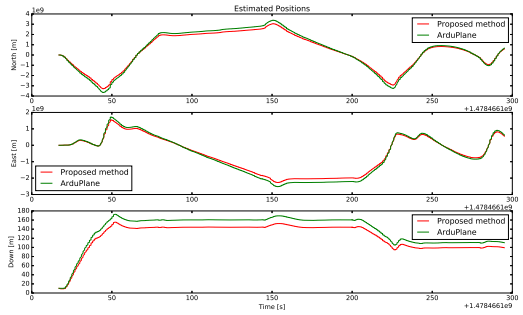
(e) Gyroscope y.



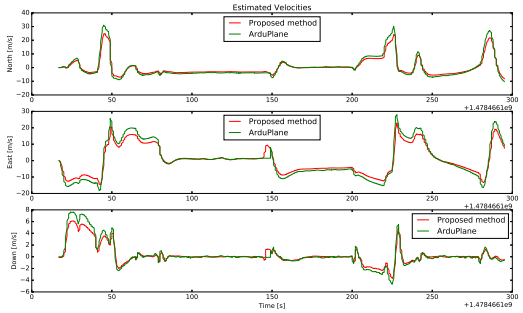
(f) Gyroscope z.

Figure 6. Measurements from two-sensor group IMU in three axis after cointegration test and residual errors.

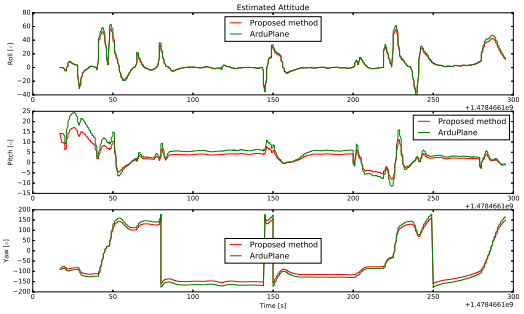
As a future work, we intend to apply the algorithm to estimate in real time the measurements of this inertial navigation system considering the operation rate of the two IMUs, that allows working with up to 800 Hz.



(a) Estimated Position.



(b) Estimated Velocity.



(c) Estimated Attitude.

Figure 7. UAV state given after apply the EKF in cointegrated time series.

ACKNOWLEDGMENT

The authors gratefully acknowledge the financial support from Brazilian Research Agencies. Edital 21-2011 Pro-Estratégia, Financiadora de Estudos e Projetos (FINEP).

REFERENCES

[1] A. C. de Carvalho Rodrigues and P. F. F. Rosa, "Fixed-wing state level hil via factor graph incremental smoothing," in *Industrial Technology (ICIT), 2015 IEEE International Conference on*. Seville: IEEE, intl.

[2] A. C. d. C. Rodrigues, "Fusão de sensores para um vant via suavização incremental baseada em grafos-fatores," Rio de Janeiro, p. 133, (2016, October 4). [Online]. Available: http://www.comp.ime.eb.br/pos/images/repositorio-dissertacoes/2015_Crivella.pdf

[3] R. A. d. R. L. R. M. E. Magalhães Neto, Jacy Montenegro; Paixão, "Uma missão de monitoramento para o projeto vant-ime: Operação região serrana-rj," in *Congresso Brasileiro de Software*, ser. AutoSOFT, no. 2. São Paulo: SBC, 2011, br.

[4] X. Wanli, B. Shuo, and L. Zhun, "The state estimation of uav based on ukf," in *Advanced Research and Technology in Industry Applications (WARTIA), 2014 IEEE Workshop on*, Sept 2014, pp. 402–405.

[5] G. Grisetti, R. Kümmerle, and K. Ni, "Robust optimization of factor graphs by using condensed measurements," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. Algarve: IEEE, intl.

[6] P. Agarwal, G. Grisetti, G. Diego Tipaldi, L. Spinello, W. Burgard, and C. Stachniss, "Experimental analysis of dynamic covariance scaling for robust map optimization under bad initial estimates," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. Hong Kong: IEEE, intl.

[7] S. Oswald, H. Kretzschmar, W. Burgard, and C. Stachniss, "Learning to give route directions from human demonstrations," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. Hong Kong: IEEE, intl.

[8] J. Shi, J. Yang, and H. Liu, "A robust and fault-tolerant filter and its application in mems-ins/gps," in *Networking, Sensing and Control (ICNSC), 2015 IEEE 12th International Conference on*, April 2015, pp. 247–251.

[9] L. Li and W. Shi, "A fault tolerant model for multi-sensor measurement," *Chinese Journal of Aeronautics*, vol. 28, no. 3, pp. 874–882, 2015.

[10] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[11] C. G. Prevost, A. Desbiens, and E. Gagnon, "Extended kalman filter for state estimation and trajectory prediction of a moving object detected by an unmanned aerial vehicle," in *2007 American Control Conference*, July 2007, pp. 1805–1810.

[12] P. Balzer, T. Trautmann, and O. Michler, "Epe and speed adaptive extended kalman filter for vehicle position and attitude estimation with low cost gnss and imu sensors," in *Informatics in Control, Automation and Robotics (ICINCO), 2014 11th International Conference on*, vol. 01, Sept 2014, pp. 649–656.

[13] K. Xiong, H. Zhang, and L. Liu, "Adaptive robust extended kalman filter for nonlinear stochastic systems," *IET Control Theory Applications*, vol. 2, no. 3, pp. 239–250, March 2008.

[14] F. Zhu, Y. Zhang, X. Su, H. Li, and H. Guo, "Gnss position estimation based on unscented kalman filter," in *2015 International Conference on Optoelectronics and Microelectronics (ICOM)*, July 2015, pp. 152–155.

[15] S. Yazdkhasti, J. Z. Sasiadek, and S. Ulrich, "Performance enhancement for gps/ins fusion by using a fuzzy adaptive unscented kalman filter," in *2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, Aug 2016, pp. 1194–1199.

[16] A. D. Team, "Arduplane," (2016, June 10). [Online]. Available: http://ardupilot.org/plane/index.html

[17] D. Titterton and J. L. Weston, *Strapdown inertial navigation technology*. IET, 2004, vol. 17.

[18] A. Nemra and N. Aouf, "Robust ins/gps sensor fusion for uav localization using sdre nonlinear filtering," *IEEE Sensors Journal*, vol. 10, no. 4, pp. 789–798, April 2010.

[19] D. A. Dickey and W. A. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *Journal of the American statistical association*, vol. 74, no. 366a, pp. 427–431, 1979.

[20] S. Johansen, "Estimation and hypothesis testing of cointegration vectors in gaussian vector autoregressive models," *Econometrica*, vol. 59, no. 6, pp. 1551–1580, (2016, June 18). [Online]. Available: http://www.jstor.org/stable/2938278