DESIGN AND IMPLEMENTATION OF SENSOR FUSION AND CONTROL FOR AN AUTONOMOUS QUADROTOR

Santiago Paternain
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
spaternain@gmail.com

Rodrigo Rosa
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
rodrigorosa.LG@gmail.com

Matías Tailanián
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
matias@tailanian.com

Rafael Canetti
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
canetti@fing.edu.uy

***Abstract*—This paper describes the design and integration of an instrumentation and control system that allows the autonomous flight of a quadrotor. A comercial frame is used, a mathematical model for the quadrotor is developed and its parameters determined from the characterization of the unit. An intelligence is integrated to play the role of the flight controller. A 9 degrees of freedom Inertial Measurement Unit (IMU) equipped with a barometer is calibrated and added to the platform. Data from the IMU is combined with the information provided by a GPS within a modified Extended Kalman Filter (EKF) to obtain a reliable estimation of the state variables. The control actions are obtained from a proportional-integral controller based on an improved Linear Quadratic Regulator (LQR) algorithm. As shown in Section (VIII), a stable autonomous platform is achieved.**

I. INTRODUCTION

A quadrotor platform can be imagined performing countless tasks, such as search and rescue, wild fire suppression, or scientific research [1]. Many applications are being developed based on such platforms, but how do they fly? An aerial vehicle is inherently unstable, staying still is not a simple task. This paper focuses on the stabilization of a quadrotor, explaining the development of the mathematical model to represent the system, the filtering techniques applied for sensor data fusion and the control system that allows the quadrotor to fly.
STATE OF THE ART

II. SYSTEM ARCHITECTURE

The platform is based on the commercial radio controlled quadrotor shown in Figure (1). The length between opposite propellers is $61.5cm$, the weight is $990g$, and it has $1300g$ of payload. The frame and the Electronic Speed Controllers (ESCs) used for the motors were preserved, whereas the IMU and intelligence were replaced by the flight controller that was developed. A BeagleBoard[1] running Linux[2] performs the computations required to convert raw data received from the IMU[3] over a UART and combine it using an EKF. The EKF overcomes the problems inherent to each sensor and filters out noise, providing a reliable estimation of the state vector. Once the current state is known, the LQR algorithm is used to derive the control actions required to bring the system to the desired set-point. A complete diagram of the implemented system is shown in Figure (2).

Fig. 1: **Commercial quadrotor acquired.**

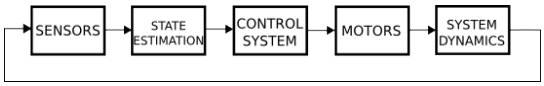


Fig. 2: **Complete system block diagram.**

The two main goals are to integrate additional sensors and intelligence to the available platform to obtain a state estimation, and design and integrate a control system that, using the state estimation, achieves the autonomous flight.

III. MODEL OF A QUADROTOR

A. Definitions

A diagram of the quadrotor is shown in Figure (3). Two of the motors rotate clockwise (2 and 4) and the other two (1 and 3) rotate counterclockwise. This configuration allows the quadrotor to rotate, tilt and gain/lose altitud by setting different

[1]BeagleBoard development board - http://beagleboard.org/
[2]Angstrom distribution: http://www.angstrom-distribution.org/
[3]*Mongoose* IMU - http://store.ckdevices.com/

speeds on each motor. Two frames of reference (Figure (3)) are constantly used through out this paper: an intertial frame $S_I - \{\hat{i}, \hat{j}, \hat{k}\}$ ($\{\vec{x}, \vec{y}, \vec{z}\}$), relative to the Earth, mapped to North, West and Up respectively, and a non-inertial frame $S_q - \{\hat{i}_q, \hat{j}_q, \hat{k}_q\}$ ($\{\vec{x}_q, \vec{y}_q, \vec{z}_q\}$) relative to the quadrotor. The mapping of one frame to the other can be achieved by applying the three rotations shown in Figure (4). The angles $\{\theta, \varphi, \psi\}$ are known as Euler angles.

B. Dynamics-kinematics of the system

From a detailed analysis of the dynamics and kinematics of the quadrotor, the equations (2) are obtained, and the state vector shown in (1) is built to describe the system at any given time. The variables with subscript q are referenced to the quadrotor frame S_q, the rest are relative to S_I:

$$\vec{X} = \{x, y, z, \theta, \varphi, \psi, v_{q_x}, v_{q_y}, v_{q_z}, \omega_{q_x}, \omega_{q_y}, \omega_{q_z}\} \quad (1)$$

where:

- $\{x, y, z\}$ represent the position of the center of mass of the system in S_I.
- $\{\theta, \varphi, \psi\}$ are the Euler angles shown in Figure (4).
- $\{v_{q_x}, v_{q_y}, v_{q_z}\}$ are the linear velocities relative to S_q.
- $\{w_{q_x}, w_{q_y}, w_{q_z}\}$ are the angular velocities relative to S_q (right hand rule applied on $\{\hat{i}_q, \hat{j}_q, \hat{k}_q\}$).

$$\begin{aligned}
\dot{x} &= v_{q_x}\cos\varphi\cos\theta + v_{q_y}(\cos\theta\sin\varphi\sin\psi - \cos\varphi\sin\theta) \\
&\quad + v_{q_z}(\sin\psi\sin\theta + \cos\psi\cos\theta\sin\varphi) \\
\dot{y} &= v_{q_x}\cos\varphi\sin\theta + v_{q_y}(\cos\psi\cos\theta + \sin\theta\sin\varphi\sin\psi) \\
&\quad + v_{q_z}(\cos\psi\sin\theta\sin\varphi - \cos\theta\sin\psi) \\
\dot{z} &= -v_{q_x}\sin\varphi + v_{q_y}\cos\varphi\sin\psi + v_{q_z}\cos\varphi\cos\psi \\
\dot{\psi} &= \omega_{q_x} + \omega_{q_z}\tan\varphi\cos\psi + \omega_{q_y}\tan\varphi\sin\psi \\
\dot{\varphi} &= \omega_{q_y}\cos\psi - \omega_{q_z}\sin\psi \\
\dot{\theta} &= \omega_{q_z}\frac{\cos\psi}{\cos\varphi} + \omega_{q_y}\frac{\sin\psi}{\cos\varphi} \\
\dot{v_{q_x}} &= v_{q_y}\omega_{q_z} - v_{q_z}\omega_{q_y} + g\sin\varphi \\
\dot{v_{q_y}} &= v_{q_z}\omega_{q_x} - v_{q_x}\omega_{q_z} - g\cos\varphi\sin\psi \\
\dot{v_{q_z}} &= v_{q_x}\omega_{q_y} - v_{q_y}\omega_{q_x} - g\cos\varphi\cos\psi + \frac{1}{M}\sum_{i=1}^{4} T_i \\
\dot{\omega_{q_x}} &= \frac{1}{I_{xx}}\omega_{q_y}\omega_{q_z}(I_{yy} - I_{zz}) \\
&\quad + \frac{1}{I_{xx}}\omega_{q_y} I_{zz_m}(\omega_1 - \omega_2 + \omega_3 - \omega_4) \\
&\quad - \frac{1}{I_{xx}} dMg\cos\varphi\sin\psi + \frac{1}{I_{xx}} L(T_2 - T_4) \\
\dot{\omega_{q_y}} &= \frac{1}{I_{yy}}\omega_{q_x}\omega_{q_z}(-I_{xx} + I_{zz}) \\
&\quad + \frac{1}{I_{yy}}\omega_{q_x} I_{zz_m}(\omega_1 - \omega_2 + \omega_3 - \omega_4) \\
&\quad - \frac{1}{I_{yy}} dMg\sin\varphi + \frac{1}{I_{yy}} L(T_3 - T_1) \\
\dot{\omega_{q_z}} &= \frac{1}{I_{zz}}(-Q_1 + Q_2 - Q_3 + Q_4)
\end{aligned} \quad (2)$$

The mathematical model developed is similar to the ones presented in [2], [3], but also takes into account that the center of gravity of the quadrotor is not at the same height as the propellers. Thus a momentum produced by the gravity force has to be added. While in [3] the linear velocities are expressed in an inertial frame, in this work are referenced to the quadrotor frame. This choice simplifies the theoretical development and the interpretation of the data provided by the IMU, which is mounted on the quadrotor and hence provides provides accelerations and angular velocities that are relative to S_q

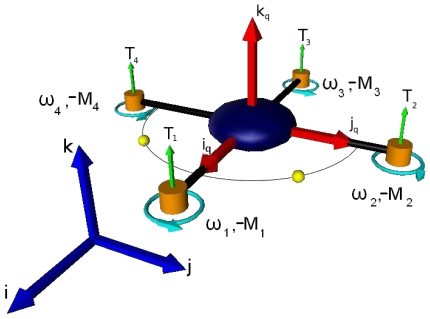
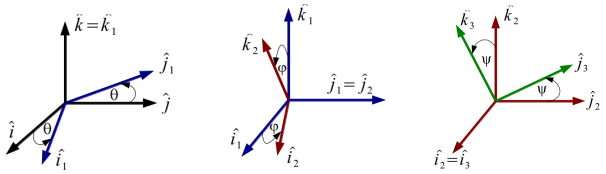


Fig. 3: **Model of the quadrotor -** The blue arrows represent the inertial reference frame S_I, and the red arrows represent the non-inertial reference frame S_q. The cyan "looped" arrows indicate the direction of rotation of each motor, which rotate at ω_i and generate a torque M_i opposite to their direction of rotation. The arrows labeled $T_{[1,2,3,4]}$ represent the thrust of the motors. The semicircle and the two yellow spheres indicate the x_q axis of the unit.



(a) Rotation 1: Axis $\hat{k}$ (b) Rotation 2: Axis $\hat{j}$ (c) Rotation 3: Axis $\hat{i}$

Fig. 4: **Mapping -** Rotations applied on S_I to obtain S_q.

IV. Sensors

In order to determine what actions should be taken, the state of the system must be known. The system uses a 9 degrees of freedom IMU and a GPS. This equipment enables direct measurement of most of the state variables. There is no direct measurement of the linear speed of the system $\{v_{q_x}, v_{q_y}, v_{q_z}\}$, so the model developed in (III) is used to estimate them.

A. IMU

The IMU is equiped with the following sensors:

- **Barometer:** Measures the absolute pressure of the environment. Variations of pressure are used to estimate variations in the altitude of the system.

- **Thermometer:** The barometer includes a thermometer. The temperature data is used to apply a temperature compensation to the calibrations performed on the gyroscope and the accelerometer.
- **Gyroscope:** A 3-axis gyroscope is used to measure angular velocity of S_q. A calibration based on [4] was designed and applied to this device. Furthermore, a temperature compensation was designed and implemented.
- **Accelerometer:** A 3-axis accelerometer is used to measure gravity. Under the hypothesis that no other accelerations are present, this allows the determination of two of the three Euler angles: $\{\psi, \phi\}$. This hypothesis is acceptable, since the accelerations involved are not significant compared to gravity. A calibration, based on [4], was designed and applied to this device, as well as a temperature-compensation.
- **Magnetometer:** In an area free of magnetic interference this 3-axis sensor will measure $\vec{B}$, the Earth's magnetic field, allowing to determine what direction is North. If the system is horizontal (or the inclination is estimated using other sensors) this sensor can be used to determine the last of the three Euler angles: θ. A calibration based on [5], [6] was performed on this sensor.

The equations used to determine the three Euler angles are the following [7]:

$$\phi = -\arcsin\left(\frac{a_x}{||\vec{a}||}\right),\ \psi = -\arctan\left(\frac{a_y}{a_z}\right)$$

$$M = \begin{pmatrix} \cos(\phi) & \sin(\phi)\sin(\psi) & \cos(\psi)\sin(\phi) \\ 0 & \cos(\psi) & -\sin(\psi) \\ -\sin(\phi) & \cos(\phi)\sin(\psi) & \cos(\phi)\cos(\psi) \end{pmatrix}$$

$$\vec{m} = M\,\vec{B},\ \theta = -\arctan\left(\frac{m[1]}{m[0]}\right)$$

B. GPS

In theory, given some initial position $\{x_0, y_0\}$ the accelerometer could be used to determine variations $\{x_0 + \Delta x, y_0 + \Delta y\}$. In practice this estimation drifts rapidly (tens of meters in a couple of seconds), so a GPS is used to determine the absolute position $\{x, y\}$ of the system, correcting the drift. The accuracy, with good sky visibility, is of 2-3 meters. The GPS's performance improves when the system is moving.

C. Sensor Specifications

Table (I) shows an outline of the specifications of the sensors used.

	Rate	Resolution
Accelerometer XY	10ms (x2)	4mg
Accelerometer Z	10ms (x2)	4mg
Gyro XY	10ms (x2)	0.07 $^o/s$
Gyro Z	10ms (x2)	0.07 $^o/s$
Barometer	10ms (x1)	1Pa
Magnetometer XY	10ms (x2)	5 mGa
Magnetometer Z	10ms (x2)	5 mGa
GPS	1s	-

TABLE I: **Sensor specifications:** A rate of "*10ms (x2)*" means that every 10ms the result of averaging 2 samples is received from the IMU.

V. Kalman Filter

In order to perform adequate control actions, a reliable estimation of the state variables must be available in real time. The Kalman Filter uses the mathematical model for the system to predict what should happen next given the current state, and corrects the prediction with the information read from the sensors, taking into consideration how much confidence is placed on the prediction and how much on the measurements. This weighted prediction-correction technique allows a smooth state estimation without the typical delay introduced by filtering, even small delays can severely affect the performance of the system.

Every sensor has its issues, but they can be combined to compensate for their limitations. The filter takes care of this by integrating all sensors in order to obtain a more accurate state estimation. Figure (5) shows a diagram of how data from the sensors is combined within the filter (EKF), assisted by the model of the system. Euler angles are primary estimated

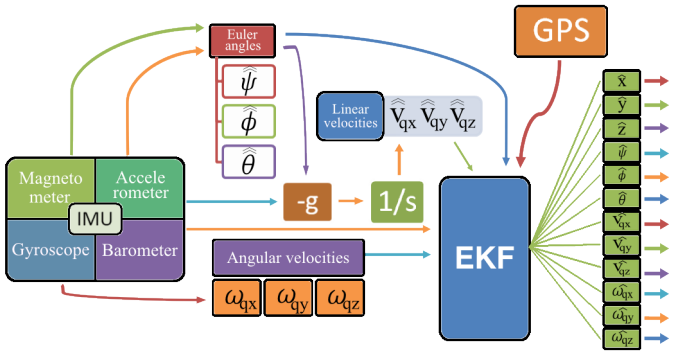


Fig. 5: **EKF -** Outline of how sensor data is combined to estimate the state variables.

from the combination of Magnetometer and Accelerometer measurements and used as one of the filter inputs. From that estimation of Euler angles and the Accelerometer measurements, the quadrotor velocities $\{v_{qx},\ v_{qy}\ v_{qz}\}$ can be deduced by subtracting gravity and an integration, which are also introduced to the filter. Angular velocities referenced to the quadrotor and height can be easily deduced from Gyroscope and Barometer measurements respectively, which are also introduced to the filter. Furthermore the GPS measurements can be also used as filter input. All this primary estimations from sensors are considered as measurements in the Kalman Filter, while the prediction is based, as said, in the dynamic model. The output of the filter is an estimation of all the variables of the state vector.

The theory behind a standard Kalman Filter does not hold for a nonlinear system. The model for the quadrotor given by (2) is highly nonlinear, so an Extended Kalman Filter (EKF)

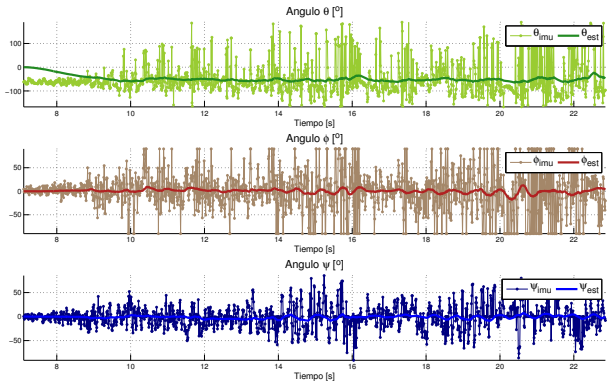


Fig. 6: **Kalman -** The three Euler angles estimation with all the motors turned on and the quadrotor in equilibrium.

is implemented. Several authors (i.e. [7], [8]) used EKF to overcome this difficulty. In this work a modified EKF was developed, similar to [7], [8], but with 3 states added representing the accelerometer bias, which improved substantially the linear velocities estimation. While the Kalman Filter ensures a statistical optimal performance, the EKF is not optimal, and it is not possible to determine the error a priori, due to high dependency of the performance with the linearization [9]. Although, EKF is the most used and popular filtering technique in navigation problems. The implementation of the EKF gave very good results, and proved to play a critical part in the system. As an example, the data provided by the accelerometer is "unusable" without filtering, and experiments with a simple low pass filter (LPF) showed that a 60ms delay introduced by the LPF severely deteriorated the performance of the system. When the EKF was assigned the task of reducing noise, the performance was significantly improved.

In each graphic of Figure (6) is shown the primary estimation obtained from sensors and the kalman estimation of that state variable for the three Euler angles. As can be seen, the noise is greatly reduced and no delay is introduced.

Two different types of data will be available, depending on the availability of GPS information. When no GPS data is available there will be no direct feedback from the sensors to correct the position on the horizontal plane, only through integration, so only the prediction step can be performed for these variables.

VI. Control Design

To work directly with equations (2), a non-linear control technique would be required. To simplifly the control system, equations (2) are linearized near certain points of operation which result in a Linear Time Invariant (LTI) system of the form:

$$\dot{\vec{X}} = A\vec{X} + B\vec{U} \tag{3}$$

This restricts the system to: Hovering[4], uniform linear trajectories, and uniform circular trajectories around the vertical axis. Working with an LTI greatly simplifies the design of the control system and does it not introduce significant limitations.

[4]Constant state vector.

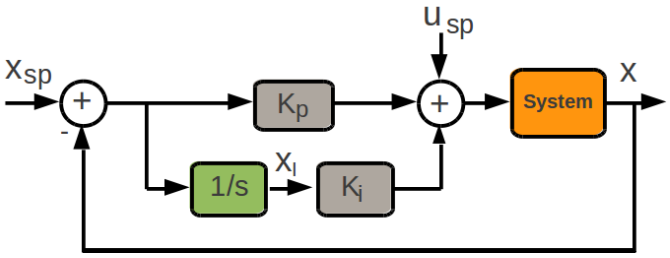


Fig. 7: **Control System:** X_{sp} and U_{sp} represent the set-points for the state vector and the speed of each motor. The K_p and K_i blocks are the proportional and integral gain matrices. The output of the system is the current state vector.

The control system is shown in Figure (7). The feedback matrices have many entries, K_p has 48 and K_i has 16. A *root locus* or *pole-zero* analysis on such a system is not a simple task, additional complications are introduced by the fact that K_p and K_i will change when the trajectory changes, they depend on the linearization of the system. A simple, automated and very commun solution is achieved by means of the LQR algorithm (see e.g.: [10]). In this work an extended state vector that includes the integrals of some state variables was introduce to take into consideration errors in the characterization and modeling of the system.

A. The LQR algorithm

A a system which is controllable and observable is guaranteed to be stable and for this type of system, the LQR algorithm provides an optimal controller (see e.g.: [11]) by minimizing the following cost function:

$$\int_0^\infty (\vec{X} - \vec{X}_{sp})' Q (\vec{X} - \vec{X}_{sp}) + (\vec{U} - \vec{U}_{sp})' R (\vec{U} - \vec{U}_{sp}) dt \tag{4}$$

where Q and R are positive-definite matrices, weighting the energy of the departure from the objective state and the objective control signal respectively. The choice of Q and R is a trade-off between the error that is tolerated and the energy that the system can use to correct it. In equation (4) $\vec{X}_{sp}$ is the objective state and $\vec{U}_{sp}$ is the theoretical control action that is compatible with the state $\vec{X}_{sp}$. Both vectors as well as $\vec{X}$ and $\vec{U}$ are time dependant.

When the system is a feedback system with known state variables, like the one shown in Figure (7), the solution is given by (5):

$$\begin{gathered} \vec{U}(t) - \vec{U}_{sp}(t) = -K(\vec{X}(t) - \vec{X}_{sp}(t)) \\ K = R^{-1} B^T P \end{gathered} \tag{5}$$

where P is the solution to Riccati's equation:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{6}$$

B. Integral effect

In this work an integral effect is added over the LQR controller. For this, an expanded state vector is defined $[X^T, X_I^T]^T$. For the controllability and observability hypothesis to hold, only some state variables may be passed through the integral control block: $\dot{X}_I = [x, y, z, \theta]^T$. After making this definitions, the system is re-written in state variables form. The method to obtain the new matrix is the same that explained in Section (VI-A)

VII. Software

A. Architecture

The software runs on several independent boards:

- **Flight controller:** Run by an ARM-Cortex-A8. Processes the data from the IMU, calculates the necessary control actions, and sets the desired speed for each motor by sending the appropriate commands, via I^2C, to the ESCs. It requires Linux specific *C* functions to handle I/O. The design of the flight controller is modular, thus replacing any part of it should be a straightforward task. The most critical aspect of the flight controller is timing, any delays will severely degrade the performance of the system.
- **IMU:** Uses an ATMega328p on the IMU. The firmware on the IMU takes care of reading from all the sensors (using I^2C) and sending a new frame of data every 10ms over a UART.
- **ESCs:** Four microprocessors[5], one for each motor.

The flight controller is accesed via an *ssh* session over *WiFi*. Once logged in, the flight controller software can be compiled and executed.

B. Implementation details

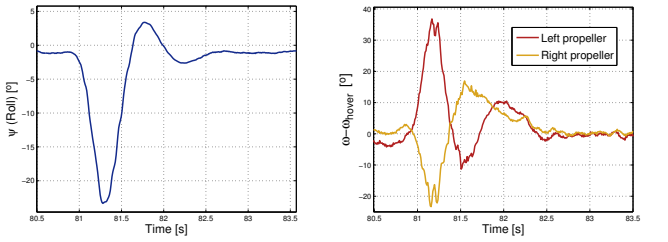
The code implements a discrete version of the LQR control system mentioned in (VI), based on a discretization of the model described in (III). There are some practical considerations worth mentioning:

- **Anti-wind-up:** In order to avoid divergency of integral terms an Anti-wind-up algorithm was implemented.
- **Limiting:** The control block sets $|\vec{X} - \vec{X}_{sp}| < |\vec{X}_{th}|$ to avoid attempts to perform actions that the system cannot handle.

VIII. Results

A. Basic stability

During this first stage, the quadrotor was only allowed to move (and correct) one of the following Euler angles: $\{\psi, \phi\}$. The control system was tuned until an acceptable behavior was obtained. This procedure was applied to each angle. Once this stage was completed the quadrotor was able to maintain horizontality by controlling $\{\psi, \phi, \omega_{q_x}, \omega_{q_y}\}$. The behavior for the Roll (ψ) angle is shown in Figure (8). While in Figure (8a) is shown the mechanical perturbation imposed to the *Roll* angle, the reaction can be observed in Figure (8b). From $t_0 = 80.9s$ to $t_1 = 81.3s$ the perturbation is present. After this, the controller is working freely. The system step response for a step of more than 20^o shows a overshoot of 3^o or 4^o and a rise time of about 0.4 s, which means an excellent performance, as can be seen in Figure (8a).



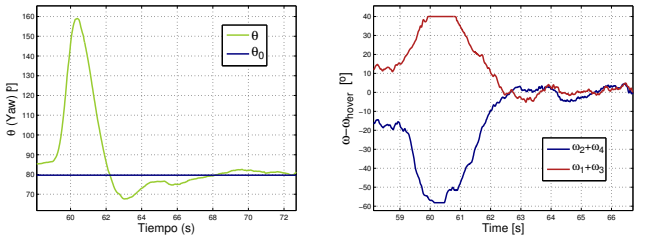
(a) Roll angle deviation

(b) Difference between ω and ω_{hover} for each motor

Fig. 8: **Stablization experiment -** The Roll angle is deviated from equilibrium and the quadrotor manage to stabilize itself.

B. Orientation

To verify orientation was correctly maintained, the speed of the motors was reduced below the level required to lift off, and the quadrotor was hung from strings. After this test, the control system was able to hold the last of the Euler angles: $\{\theta\}$. In the Figure (9) is shown the system behavior against a mechanical perturbation (from $t_0 = 59.3s$ to $t_1 = 60.5s$) in yaw angle. To rotate, the system must generate a torque in the opposite direction of the perturbation by unbalancing the angular velocity of the motors in pairs: ω_1 and ω_3 in one direction and ω_2 and ω_4 in the other. In this case, the rise time is of about 1.6 s, much higher than in the other two Euler angles, which can be permitted as Yaw angle is not such critical for stability. The overshoot obtained is of about 13% of the step size.



(a) Yaw angle deviation

(b) Difference between ω and ω_{hover} for each pair of motors.

Fig. 9: **Orientation experiment -** The Yaw angle is deviated from equilibrium and the quadrotor manage to stabilize itself.

C. Limited Flying

At this point the three Euler angles are controlled, so a basic level of stability is expected, and the quadrotor is allowed to fly. After this test, the control system was able to take-off and to hold altitud and vertical speed: $\{z, v_{q_z}\}$.

[5]C8051F330/1 - Silicon Labs.

Figure (10) shows altitude during takeoff from altitude 0m until a target altitude of 1m is achieved. As can be seen, the system presents an overshoot of about a meter, which is considered as acceptable for many applications.

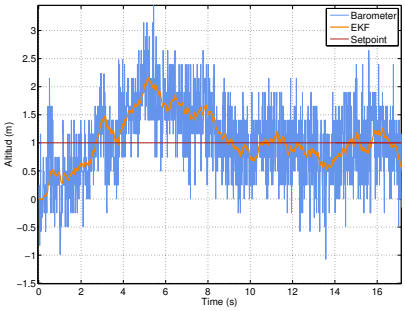


Fig. 10: **Altitud:** Performance during takeoff (from 0m to 1m).

IX. Conclusion

The final plataform implemented is shown in Figure (11). An acrylic box was added in the bottom of the quadrotor to hold the added intelligence. The GPS is attached at the upper part of the quadrotor, so as it has the best satellite visibility. Aluminum protections were added around all motors and propellers so as damages can be prevented.

The two main goals were successfuly achieved. First, the design and implementation of the sensor fusion was accomplished by the integration of all the measurements in a modified Extended Kalman Filter which is used for denoising purposes without introducing delay. In addition a basic controller, capable of stabilizing the system, was implemented based on clasic LQR algorithm and adding extra states to perform also an integral control action. A simplified dynamic model was derived as a first step for both Kalman Filter and controller.

The Extended Kalman Filter implemented gave very good results for all the state variables except for the linear velocities on axes $\{i_q,\ j_q\}$. The main reason is that a velocity sensor is not available, thus, it can only be infered by integrating the accelerations. Therefore, an extra sensor should be used (i.e. a more accurate GPS) or a more sophisticated algorithm should be develped.

The controller was successfully tested for attitude and height. Due to lack of precision of mentioned velocities this information is not used. However, several simulations were made assuming that a proper velocity estimation is available and the designed controller proved to work properly.

An experimental platform for future research was design and built, its modular design is a very important feature in this perspective, since the replacement of any part is a straightforward task.

Fig. 11: **Final setup -** quadrotor fully equipped whith aluminum protections

References

[1] Z. Sarris, "Survey of uav applications in civil markets," STN ATLAS-3Sigma AE and Technical University of Crete, June 2001.

[2] T. Bresciani, "Modeling, identification and control of a quadrotor helicopter," Master's thesis, Lund University, October 2008.

[3] M. Vendittelli, "Quadrotor modeling." Course: "Elective in robotics", Sapienza Universit Di Roma, November 2011.

[4] I. Skog and P. Händel, "Calibration of a mems inertial measurement unit," in *XVII Imeko World Congress, Metrology for a Sustainable Development*, (Rio de Janeiro), September 2006.

[5] C. Konvalin, "Compensating for tilt, hard iron and soft iron effects," Tech. Rep. MTD-0802, Memsense, August 2008.

[6] A. Barraud, "Magnetometers calibration." http://www.mathworks.com/matlabcentral/fileexchange/23398-magnetometers-calibration, March 2009.

[7] H. Zhao and Z. Wang, "Motion measurement using inertial sensors, ultrasonic sensors, and magnetometers with extended kalman filter for data fusion," *Sensors Journal, IEEE*, vol. 12, pp. 943–953, May 2012.

[8] L. Tams, G. Lazea, R. Robotin, C. Marcu, S. Herle, and Z. Szekely, "State estimation based on kalman filtering techniques in navigation," in *IEEE International Conference on Automation, Quality and Testing, Robotics*, pp. 147–152, IEEE, May 2008.

[9] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, vol. 1 of *Prentice Hall singal processing*. Pearson, 2 ed., 2011.

[10] A. F. Soslashrensen, "Autonomous control of a miniature quadrotor following fast trajectories," Master's thesis, Aalborg University/U.C. Berkeley, 2009-2010.

[11] J. P.Hespanha, "Undergraduate lecture notes on lqg/lqr controller design," 2007.