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, sensor fusion 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. Sensor fusion is done by two modified Extended Kalman Filters (EKF): one combining data provided by IMU and the other also including the information provided by GPS. A reliable estimation of the state variables is obtained. Three states representing systematic bias in the accelerometer measurements are also added to the EKF, which improves the inertial estimation of the position. The control actions are obtained from a proportional-integral controller based on an improved Linear Quadratic Regulator (LQR) algorithm. A stable autonomous platform is achieved.**

I. INTRODUCTION

There is an important growth in the interest about unmanned aerial vehicles (UAV) in relation to its capabilities to perform a wide spectrum of tasks as monitoring, surveillance, aerial photography, exploration, delivery, rescue, remote sensing, etc. The continuous advances in low power MEMS sensor devices, low power embedded processors, energy storage through efficient electric batteries and electrical motors technologies have been driving forces in the process of implementation of new and more sophisticated devices of this kind, particularly miniature flying robots. Quadrotors UAV is one of the most popular architectures. They emerged also as a typical platform for research. There is much research work about path planning and vision towards navigation in unstructured scenarios (e.g.: [1]–[3]), fault tolerant navigation (e.g.: [4]), path tracking (e.g.: [5]), control techniques for UAV such as neural networks [6], gain-schedulling [4], feedback linearization (e.g.: [3], [7]), PI control [5], Adaptive Control [8], LQR (e.g.: [2], [9]), backstepping control (e.g.: [1], [3]), etc. If state estimation, path planning or control techniques are to be experimented in a physical setup, an important issue is to have the possibility to gain access to the different blocks of the navigational system. The purpose of the reported work is to develop the instrumentation and control of an autonomous quadrotor in which is possible to modify and test different state estimation techniques, path planning algorithms and control laws. The solution was reached using a commercial mechanical platform, and design and build an ad-hoc instrumentation and control navigation system. 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.

Fig. 1: **Commercial motorized-frame used.**

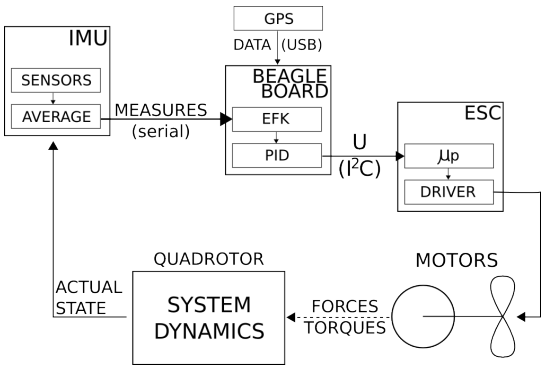


Fig. 2: **General diagram.**

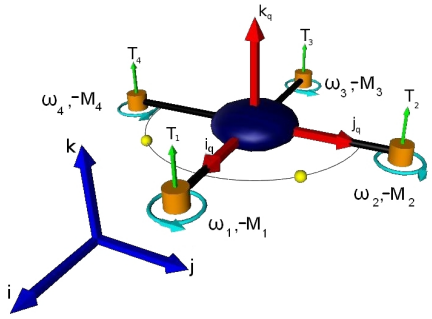


Fig. 3: **Model of the quadrotor -** The blue arrows represent the inertial reference frame S_I, and the red arrows represent the non-inertial reference frame S_q. The cyan "looped" arrows indicate the direction of rotation of each motor, which rotate at ω_i and generate a torque M_i opposite to their 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.

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 $990g$ (including battery), and it has $1300g$ of payload. The frame, motors 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). 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 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.

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

B. Dynamics-kinematics of the system

From a detailed analysis of the dynamics and kinematics of the quadrotor, the equations (1) are obtained, and the state vector shown in (2) 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:

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

$$
\mathbf{x} = \{x, y, z, \theta, \varphi, \psi, v_{q_z}, v_{q_y}, v_{q_z}, \omega_{q_x}, \omega_{q_y}, \omega_{q_z}\} \tag{2}
$$

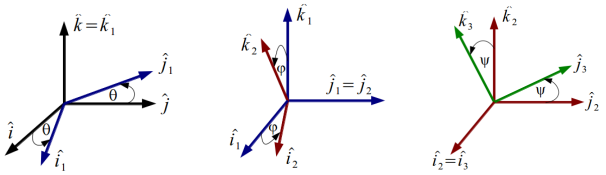
where:

- $\{x, y, z\}$ 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 $\{\hat{i}_q, \hat{j}_q, \hat{k}_q\}$).
- $T_1(\omega_1)$, $T_2(\omega_2)$, $T_3(\omega_3)$, $T_4(\omega_4)$ are the thrust of the motors.

The dynamical model considered, derived from equations 1, is

$$
\dot{\mathbf{x}} = \mathcal{F}(\mathbf{x}, \mathbf{u}) \tag{3}
$$

where $\mathbf{u} = \{\omega_1, \omega_2, \omega_3, \omega_4\}$ is the controllable input of the system.



(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.

The mathematical model developed is similar to the ones presented in [10], [11], 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 [11] 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 accelerations and angular velocities that are relative to 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 $\{v_{q_z}, v_{q_y}, v_{q_z}\}$, 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. The reader can find the details in this procedure in [12].
- **Gyroscope:** A 3-axis gyroscope is used to measure angular velocity of S_q. A calibration based on [13] 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 [13], 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 $\mathbf{B}$, 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 [14], [15] was performed on this sensor.

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 (ten meters in about ten 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. It 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, the accelerometer drifts over time; the gyroscope is very sensible to the vibrations generated by the motors; the magnetometer measure is distorted by ferromagnetic materials; the GPS has a very poor accuracy and a slow update rate. Each sensor by itself is very limited, 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.

The theory behind a standard Kalman Filter does not hold for a nonlinear system. The model for the quadrotor given by (1) is highly nonlinear, so an Extended Kalman Filter (EKF) is implemented. Several authors (e.g. [16], [17]) used EKF to overcome this difficulty. In this work a modified EKF was developed, similar to [16], [17], but with 3 states added representing the accelerometer bias, which improved substantially

the linear velocities and position 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 [18]. Although, EKF is the most used and popular filtering technique in navigation problems.

A. Mathematical model

Let us define an extended state vector $\mathbf{x}^e$, similar to the described in equation (2), but with 3 added states $\{a_x^b, a_y^b, a_z^b\}$ representing an estimation of the systematic error introduced by the accelerometer. Let $\mathbf{x}_k^e$ be the extended state vector estimation at time k

$$\mathbf{x}_k^e = \{x, y, z, \theta, \varphi, \psi, v_{q_z}, v_{q_y}, v_{q_z}, \omega_{q_x}, \omega_{q_y}, \omega_{q_z}, a_x^b, a_y^b, a_z^b\}$$

$\boldsymbol{\eta}_k^w$ and $\boldsymbol{\eta}_k^v$ are the process and observation noises, which are both assumed to be zero mean multivariate Gaussian noises with covariance $\mathbf{Q}_k$ and $\mathbf{R}_k$ respectively.

The dynamic system follows the model:

$$\mathbf{x}_k^e = f(\mathbf{x}_{k-1}^e, \mathbf{u}_{k-1}) + \boldsymbol{\eta}_{k-1}^w \quad (4)$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \boldsymbol{\eta}_k^v \quad (5)$$

where $\mathbf{z}_k$ is the observation at time k. The function f, based on the dynamics of the system (equations (1)), is deduced form equation (3) and is used to compute the predicted state from the previous one. The function h is used to compute the predicted measurement from the predicted state. In other words, f keeps the information about state evolution, and h represents the transformation between the state vector and the ideal (noiseless) observation. The state transition and observation models do not need to be linear functions, but must be differentiable. f and h cannot be directly applied to the covariance. Instead a partial differential matrix (the *Jacobian*) is computed, which is evaluated in the current state at each time step. Due to dynamical linearization, the EKF cannot ensure statistic optimality, because the precision highly depends on the linearization precision [18].

B. Implementation

The implementation of the modified EKF uses the inertial measurements obtained from the IMU and integrates the position information provided by the GPS. Due to the very slow sampling rate of GPS (approximately 1Hz), and the need of taking a new control action much faster, two different EKFs are implemented; the first one using only the inertial measurements (EKF_{IMU}) while there is no GPS information, and the other one to be used when a new GPS sample is available ($EKF_{IMU+GPS}$). The prediction and update equations are presented below.

1) Prediction and update equations for EKF_{IMU}:

- Prediction:
  - $\hat{\mathbf{x}}_{k|k-1}^e = f(\hat{\mathbf{x}}_{k-1|k-1}^e, \mathbf{u}_{k-1})$
  - $\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}$
- Update
  - $\tilde{\mathbf{y}}_k = \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}^e)$
  - $\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^\top + \mathbf{R}_k$
  - $\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^\top\mathbf{S}_k^{-1}$
  - $\hat{\mathbf{x}}_{k|k}^e = \hat{\mathbf{x}}_{k|k-1}^e + \mathbf{K}_k\tilde{\mathbf{y}}_k$
  - $\mathbf{P}_{k|k} = (I - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}$

where $\mathbf{z}_k$ is the observation at time k.
The state and observation transition matrices are

$$\mathbf{F}_{k-1} = \left.\frac{\partial f}{\partial \mathbf{x}^e}\right|_{\hat{\mathbf{x}}_{k-1|k-1}^e, \mathbf{u}_{k-1}}, \qquad \mathbf{H}_k = \left.\frac{\partial h}{\partial \mathbf{x}^e}\right|_{\hat{\mathbf{x}}_{k|k-1}^e}$$

2) Prediction and update equations for $EKF_{IMU+GPS}$:

- Prediction:
  - $\hat{\mathbf{x}}_{k|k-1}^e = f_G(\hat{\mathbf{x}}_{k-1|k-1}^e, \mathbf{u}_{k-1})$
  - $\mathbf{P}_{\mathbf{G}k|k-1} = \mathbf{F}_{\mathbf{G}k-1}\mathbf{P}_{\mathbf{G}k-1|k-1}\mathbf{F}_{\mathbf{G}k-1}^T + \mathbf{Q}_{\mathbf{G}k-1}$
- Update
  - $\tilde{\mathbf{y}_{\mathbf{G}k}} = \mathbf{z}_{\mathbf{G}k} - h_G(\hat{\mathbf{x}}_{k|k-1}^e)$
  - $\mathbf{S}_{\mathbf{G}k} = \mathbf{H}_{\mathbf{G}k}\mathbf{P}_{\mathbf{G}k|k-1}\mathbf{H}_{\mathbf{G}k}^\top + \mathbf{R}_{\mathbf{G}k}$
  - $\mathbf{K}_{\mathbf{G}k} = \mathbf{P}_{\mathbf{G}k|k-1}\mathbf{H}_{\mathbf{G}k}^\top\mathbf{S}_{\mathbf{G}k}^{-1}$
  - $\hat{\mathbf{x}}_{k|k}^e = \hat{\mathbf{x}}_{k|k-1}^e + \mathbf{K}_{\mathbf{G}k}\tilde{\mathbf{y}}_k$
  - $\mathbf{P}_{\mathbf{G}k|k} = (I - \mathbf{K}_{\mathbf{G}k}\mathbf{H}_{\mathbf{G}k})\mathbf{P}_{k|k-1}$

The two sets of prediction and update equations, for EKF_{IMU} and $EKF_{IMU+GPS}$ are very similar, and the subindex $_G$ is used to explicitly show that the variables are different in each filter. This difference is caused because $\mathbf{z}_k$ and $\mathbf{z}_{\mathbf{G}k}$ are different since $\mathbf{z}_{\mathbf{G}k}$ includes GPS data. Note that $\hat{\mathbf{x}}_{k|k-1}^e$ is always the same.

This two filters are in fact very similar, but while in the EKF_{IMU} the position is estimated by the prediction based on the dynamics-kinematics of the system, in the $EKF_{IMU+GPS}$ the samples from GPS are used as feedback for the position estimation.

The position estimation without using GPS is very poor, because of the accumulated error produced by the double integration of the accelerometer measurement. The EKF_{IMU} is meant to keep a reasonable estimation of position while the system is waiting for a new GPS sample. $EKF_{IMU+GPS}$ gives a high weight to the GPS measurement so it can be used as a correction measurement, avoiding the drift that the integration may cause.

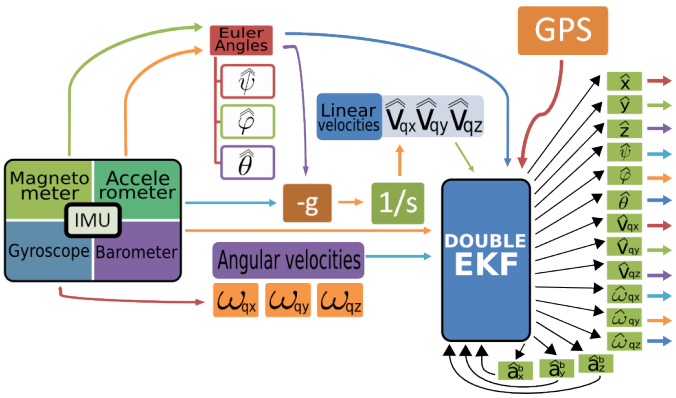
Figure (5) shows a diagram of how data from the sensors is combined within the filter (EKF), assisted by the model of the system.

Euler angles are primary estimated from the combination of Magnetometer and Accelerometer measurements as follows [16]:

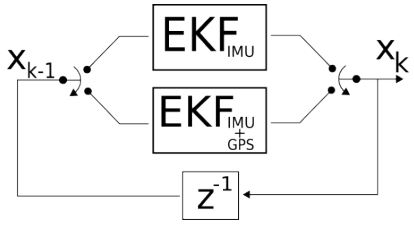
$$\phi = -\arcsin\left(\frac{a_x}{||\mathbf{a}||}\right), \quad \psi = -\arctan\left(\frac{a_y}{a_z}\right)$$

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

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



(a) **General diagram**



(b) **Double EKF -** Detail.

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

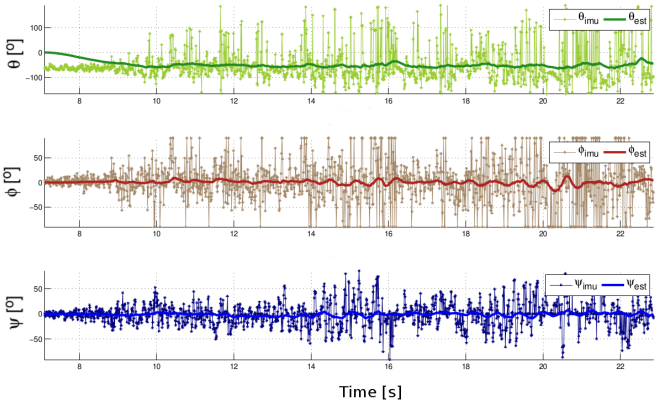
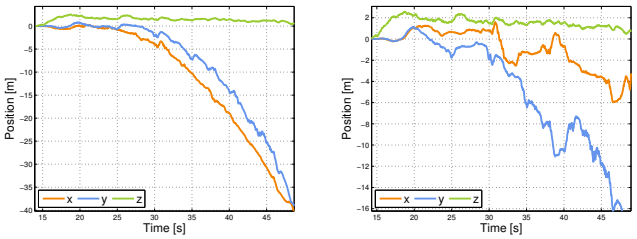


Fig. 6: **Kalman -** The three Euler angles estimation with all the motors turned on and the quadrotor in equilibrium.



(a) Without bias estimation (b) With bias estimation

Fig. 7: **Comparison -** Position estimation.

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 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.

C. Results

The implementation of the double 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 deteriorates 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.

Although the accelerometer is calibrated considering axis non-orthogonality and temperature dependency, there are some other effects that may affect the measurement, such as humidity, nonlinearities or some not considered interference. For estimating the position, the acceleration is integrated twice, so every error is propagated and may become considerable. In the calibration procedure [12], there can be some error while ensuring horizontality that causes a systematic error in the measurements. The three added states $\{a_x^b, a_y^b, a_z^b\}$ keeps an estimation of the systematic error induced by the accelerometer and improves significantly the estimation of the linear velocities and position. In Figure (7) is shown the position estimation in a real flight using only EKF_{IMU} with the bias estimation (Figure (7b)), and without using it (Figure (7a)). Although not having a ground truth for the position, it is clear that the position drift is considerably reduced. Taking into account that the flight was performed in a 10m square room, the real movement is bounded by the room size.

As said, two different types of data will be available, depending on the availability of GPS information. When GPS data is available, $EKF_{IMU+GPS}$ shows up and gives feedback to position estimation. Figure (8) shows the results of estimating the position using the combination of the two EKFs described. The same real flight data is used for this estimation, but simulated GPS data is added. As can be seen, GPS data is available at 1Hz, and the position is improved.

VI. Control Design

To work directly with equations (1), a non-linear control technique would be required. To simplify the control system, equations (1) are linearized near certain points of operation which result in a Linear Time Invariant (LTI) system of the

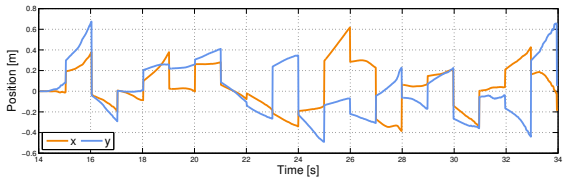


Fig. 8: **Position estimation -** Using GPS.

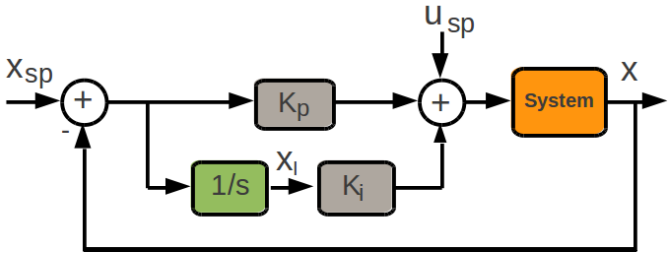


Fig. 9: **Control System:** X_{sp} and U_{sp} represent the set-points for the state vector and the speed of each motor. The K_p and K_i blocks are the proportional and integral gain matrices. The output of the system is the current state vector.

form:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \tag{6}$$

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.

The control system is shown in Figure (9). 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.: [19]). 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.: [9]) by minimizing the following cost function:

$$\int_0^\infty (\mathbf{x}-\mathbf{x}_{sp})^T\mathbf{Q}(\mathbf{x}-\mathbf{x}_{sp}) + (\mathbf{u}-\mathbf{u}_{sp})^T\mathbf{R}(\mathbf{u}-\mathbf{u}_{sp})dt \tag{7}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are positive-definite matrices, weighting the energy of the departure from the objective state and the objective control signal respectively. The choice of $\mathbf{Q}$ and $\mathbf{R}$ is a trade-off between the error that is tolerated and the energy that the system can use to correct it. In equation (7) $\mathbf{x}_{sp}$ is the objective state and $\mathbf{u}_{sp}$ is the theoretical control action that is compatible with the state $\mathbf{x}_{sp}$. Both vectors as well as $\mathbf{x}$ and $\mathbf{u}$ are time dependent.

When the system is a feedback system with known state variables, like the one shown in Figure (9), the solution is given by (8):

$$\begin{gathered}\mathbf{u}(t) - \mathbf{u}_{sp}(t) = -K(\mathbf{x}(t) - \mathbf{x}_{sp}(t)) \\ \mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}\end{gathered} \tag{8}$$

where $\mathbf{P}$ is the solution to Riccati's equation:

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = 0 \tag{9}$$

B. Integral effect

In this work an integral effect is added over the LQR controller. For this, an expanded state vector is defined $[\mathbf{x}^T, \mathbf{x}_I^T]^T$. For the controllability and observability hypothesis to hold, only some state variables may be passed through the integral control block: $\dot{\mathbf{x}}_I = [x, y, z, \theta]^T$. 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.

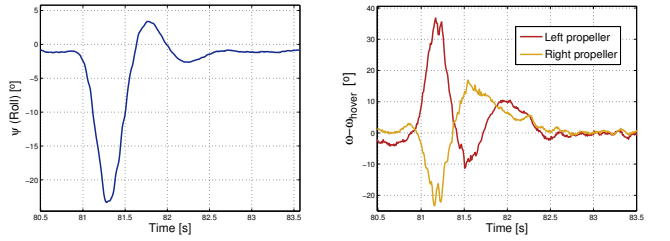
[4]Constant state vector.

[5]C8051F330/1 - Silicon Labs.

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 (10). While in Figure (10a) is shown the mechanical perturbation imposed to the *Roll* angle, the reaction can be observed in Figure (10b). 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 (10a).



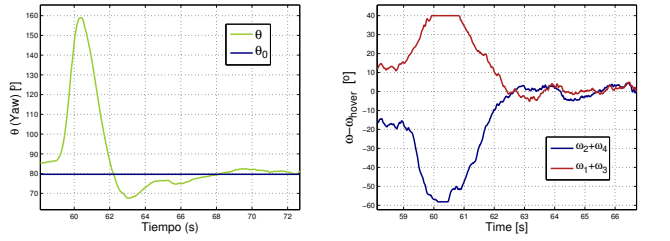
(a) Roll angle deviation

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

Fig. 10: **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 (11) is shown the system behavior against



(a) Yaw angle deviation

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

Fig. 11: **Orientation experiment -** The Yaw angle is deviated from equilibrium and the quadrotor manage to stabilize itself.

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.

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: $\{z, v_{q_z}\}$.

Figure (12) 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.

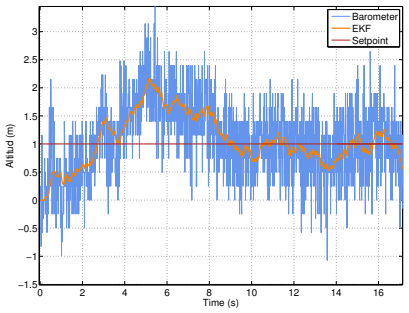


Fig. 12: **Altitud:** Performance during takeoff (from 0m to 1m).

IX. Conclusion

The final plataform implemented is shown in Figure (13). 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. 13: **Final setup -** quadrotor fully equipped whith aluminum protections

REFERENCES

[1] G. Qingbo, S. Huan, and H. Qiong, "Obstacle avoidance approaches for quadrotor uav based on backstepping technique," Control and Decision Conference (CCDC), vol. ., pp. 3613–3617, May 2013.

[2] J. Park and Y. Kim, "Stereo vision based collision avoidance of quadrotor uav," Control, Automation and Systems (ICCAS), vol. ., pp. 173–178, Oct 2012.

[3] E. Altug, J. Ostrowski, and R. Mahony, "Control of a quadrotor helicopter using visual feedback," ICRA '02. IEEE International Conference, vol. 1, pp. 72–77, . 2002.

[4] I. Sadeghzadeh, A. Mehta, A. Chamseddine, and Y. Zhang, "Active fault tolerant control of a quadrotor uav based on gainscheduled pid control," Electrical & Computer Engineering (CCECE), vol. ., pp. 1–4, May 2012.

[5] A. Puls, T.; Hein, "3d trajectory control for quadrocopter," *Intelligent Robots and Systems (IROS)*, vol. 1, pp. 640–645, Oct 2010.

[6] S. Dierks, T.; Jagannathan, "Output feedback control of a quadrotor uav using neural networks," *Neural Networks, IEEE Transactions*, vol. 21, pp. 50–66, Jan 2010.

[7] A. Mokhtari, A.; Benallegue, "Dynamic feedback controller of euler angles and wind parameters estimation for a quadrotor unmanned aerial vehicle," ICRA '04. IEEE International Conference, vol. 3, pp. 2359–2366, April 2004.

[8] A. Mohammadi, M.; Mohammad Shahri, "Modelling and decentralized adaptive tracking control of a quadrotor uav," *Robotics and Mechatronics (ICRoM)*, vol. ., pp. 293–300, Feb 2013.

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

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

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

[12] S. Paternain, M. Tailanin, and R. Canetti, "Calibration of an inertial measurement unit," *ICAR. IEEE International Conference*, 2013.

[13] 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.

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

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

[16] 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.

[17] 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.

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

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