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 acquired, a mathematical model for the quadrotor is developed and its parameters determined from the characterization of the unit. An external 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 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 the LQR algorithm. As shown in section (VII-A), 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 [2] [3] to represent the system, the filtering techniques applied for sensor data fusion [4] [5] and the control system [2] [6] that allows the quadrotor to fly. The platform is based on the commercial radio controlled quadrotor, shown in figure (1). The length between opposite propelers is $61.5\ cm$, the wheight is $990\ g$, and it has $1300\ g$ of payload. The frame and the electronic speed controllers (ESCs) used for the motores 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 requiered 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 [6] [7] is used to derive the control actions required to bring the system to the desired setpoint.

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

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.

II. MODEL OF A QUADROTOR

A. Definitions

A diagram of the quadrotor is shown in figure (2). 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 (2)) 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-intertial 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 (6). 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

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

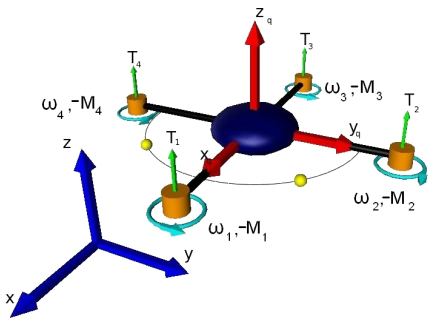


Fig. 2: **Model of the quadrotor -** The blue arrows represent the intertial reference frame S_I, and the red arrows represent the non-inertial reference frame S_q. The cyan "looped" arrows indicate the direction of rotation of each motor, which rotate at ω_i and generate a torque M_i opposite to their 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.

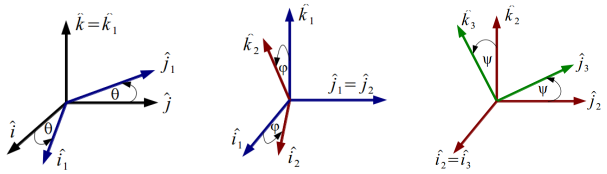
given time. The variables with subscript q are referenced to the quadrotor frame S_q, the rest are relative to S_I. This choice simplifies the theoretical development and the interpretation of the data provided by the IMU, which is mounted on the quadrotor and hence provides provides accelerations and angular velocities that are relative to S_q:

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

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 (6).
- $\{v_{q_x}, v_{q_y}, v_{q_z}\}$ 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 $\{\hat{i}_q, \hat{j}_q, \hat{k}_q\}$).

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



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

Fig. 3: **Mapping -** Rotations applied on S_I to obtain S_q.

III. 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 (II) 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 altitud 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 [8], was designed and applied to this device, as well as a temperature-compensation.
- **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 [8], 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 [9] and [10] was performed on this sensor.

The equations used to determine the three Euler angles are the following [4]:

$$\phi = -\arcsin\left(\frac{Acc_x}{||\vec{Acc}||}\right)$$

$$\psi = -\arctan\left(\frac{Acc_y}{Acc_z}\right)$$

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

$$\vec{v} = M.\frac{\vec{B}}{||\vec{B}||}$$

$$\theta = -\arctan\left(\frac{v[1]}{v[0]}\right)$$

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.

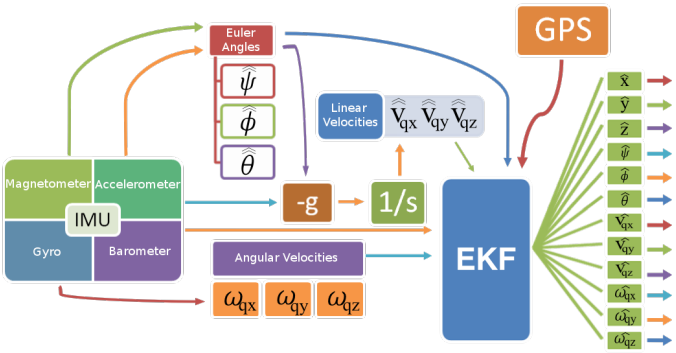


Fig. 4: **EKF -** Outline of how sensor data is combined to estimate the state variables.

IV. 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 severly affect the performance of the system.

Every sensor has its issues: The gyro drifts over time; the accelerometer is very sensible to the vibrations generated by the motors; the magnetometer 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.

A. EKF

The theory behind a standard Kalman Filter does not hold for a system that is not linear. The model for the quadrotor given by (2) is highly non-linear, so an EKF is implemented based on [4] and [5]. An EKF is compatible with non-linear systems, but it is not optimal, the performance is highly dependant on the linearizacion [11]. 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 severly deteriorated the performance of the system. When the EKF was assigned the task of reducing noise, the perfomance was significatively improved.

B. Integration with the system

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.

Figure (4) shows a diagram of how data from the sensors is combined within the EKF, assisted by the model of the system. The output of the filter is an estimation the all the variables of the state vector.

V. 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 control system and does it not introduce significant limitations.

The control system is shown in figure ((5)). 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 trayectory changes, they depend on the linearization of the system. A simple and automated solution is achieved by means of the LQR algorithm. The initial design did not include the K_i term, it was introduced to take into consideration errors in the characterization and modelling 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 [6] by minimizing the following cost function:

$$\int_0^\infty \vec{X}'(t)Q\vec{X}(t) + \vec{U}'(t)R\vec{U}(t)dt \tag{4}$$

where Q and R are positive-definite matrices, representing the energy of the controlled states and of the control signal, respectively. The choice of Q and R is a tradeoff between the error that is tolerated and the energy that the system can use to correct it.

[4]Constant state vector.

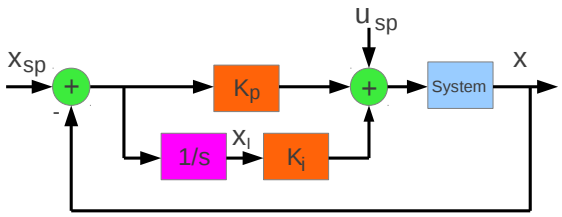


Fig. 5: **Control System:** x_{sp} and u_{sp} represent the setpoints for the state vector and the speed of each motor. The K_p and K_i blocks are the proportional and integral gain matrices. The output of the system is the current state vector.

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

$$\begin{gathered}\vec{U}(t) - \vec{U}^*(t) = -K(\vec{X}(t) - \vec{X}^*(t)) \\ K = R^{-1}B^TP\end{gathered} \tag{5}$$

where P is the solution to Riccati's equation:

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

For the controllability and observability hypothesis to hold, only some state variables may be passed through the integral control block: $\{x, y, z, \theta\}$.

VI. Software

A. Architecture

The software runs on several independant 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. It was developed in modules, 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 code running on the ESCs is not available, but the communication protocol was decoded using a logic analyzer.

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

[5]C8051F330/1 - Silicon Labs.

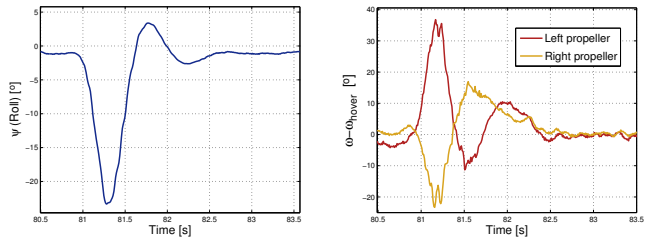
B. Implementation details

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

- **Saturation:** The integral term saturates it is designed to compensate for small errors in characterization/modelling and should never grow too much. Large diferences between the current state and the setpoint should be handled by the proportional term.
- **Limiting:** The control block sets $|\vec{X} - \vec{X}_{sp}| < |\vec{X}_{th}|$ to avoid attempts to perform actions that the system cannot handle.

VII. Results

A. Tests



(a) Roll angle deviation

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

Fig. 6: **Stablization experiment -** The Roll angle is deviated from equilibrium and the quadrotor manage to stabilize itself.

The testing phase was separated into several parts:

1) **Basic Stability**: During this first stage, the quadcopter was only allowed to move (and correct) one of the 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 quadcopter was able to mantain horizontality by controlling $\{\psi, \phi, \omega_{q_x}, \omega_{q_y}\}$.
2) **Orientation**: To verify orientation was correctly maintained the speed of the motors was reduced, below the level required to lift off, and the quadcopter was hung from strings. During this test offsets can be set on each motors to compensate for differences between them. After this test, the control system was able to hold the last of the Euler angles: $\{\theta, \omega_{q_z}\}$.
3) **Limited flying**: At this point the three Euler angles are controlled, so a basic level of stability is expected. During this stage strings were attached to the quadcopter from the top and bottom, and it was allowed to fly. After this test, the control system was able to take-off and to hold altitud and vertical speed: $\{z, v_{q_z}\}$.
4) **Free flight**: The last step was to let the quadcopter fly. This stage required the GPS. After it was completed, the control system was able to hold it's horizontal position $\{x, y, v_{q_x}, v_{q_y}\}$, completing control over the state variables.

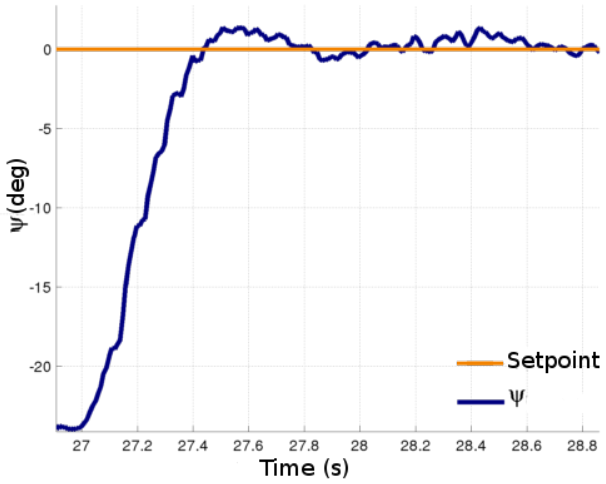


Fig. 7: **Step response:** Performance during an experiment limited to a single degree of freedom (ψ).

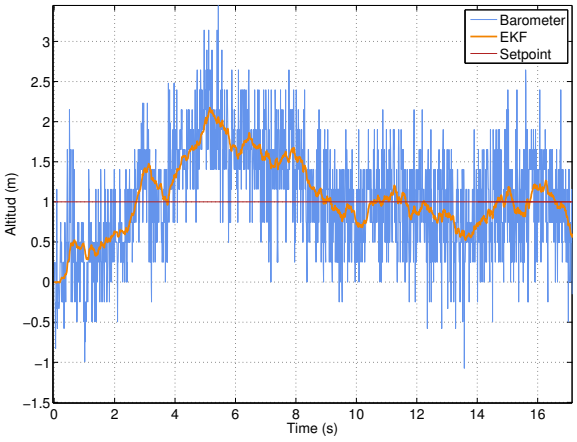


Fig. 8: **Altitud:** Performance during takeoff (from 0m to 1m).

Free flight tests were limited due to the poor reliabitily of the ESCs provided by LotusRC[6], which showed issues such as overheating until burning out and random glitches that would cause motors to turn off.

B. System response

To ilustrate some results, figure (7) shows the step response for ψ during an experiment where this was the only degree of freedom. Figure (8) shows altitud during takeoff from altitud 0m until a target altitud of 1m.

Flight test were satisfactory in windless environments. In presence of wind the system is not stable. The aluminium protections added (to protect the motors from the faulty ESCs) have flat surfaces that may intensify the effect of the wind.

[6]http://www.lotusrc.com/

VIII. Conclusion

The main goal of the project was achieved: A quadcopter able to fly on its own. The system was successfully tested. A basic control was implemented, capable of stabilizing the system. The doors are open for future projects to add additional levels of intelligence allowing the quadcopter to perform tasks such as power line inspection, surveillance, assitance during natural disasters, etc.

A. Future Work

- The LotusRC ESCs proved to be extremely unreliable[7] and severly delayed the project. Replacing the ESCs is a critical task.
- The choice of implementing the flight controller on an operating system that does not run in real-time made timing a complicated issue. These issues were dealt with, but assinging additional tasks to the system will surely distrupt timing. Porting the stabilization system, which requires a very fast and steady loop, to a dedicated microprocessor would be a significant improvement. It would allow the code running on top of the operating system to perform higher level tasks, such as Visual Simultaneous Localization and Mapping (VSLAM), navigation, tracking, etc.
- The fact that the system does not work in a windy environment severly limits the range of possible applications, solving this issue by either reducing the aerodynamic resistance or implementing a different control system would be a significant improvement.

Acknowledgment

The authors would like to thank family and friends for their support during this year, and the staff of the University for their commitment to the project.

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] 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.

[5] 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.

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

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

[8] 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.

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

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

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

[7] Several versions were tested, all of then showed the same issues.