IMPLEMENTATION OF A QUADROTOR UAV

Santiago Paternain
Faculty of Engineering
University of the Republic
Montevideo, Uruguay
spaternain@gmail.com

Rodrigo Rosa
Faculty of Engineering
University of the Republic
Montevideo, Uruguay
rodrigorosa.LG@gmail.com

Matías Tailanián
Faculty of Engineering
University of the Republic
Montevideo, Uruguay
matias@tailanian.com

Rafael Canetti
Faculty of Engineering
University of the Republic
Montevideo, Uruguay
canetti@anii.org.uy

***Abstract*—This paper describes the design and integration of a control system that allows the autonomous flight of a commercial quadrotor. A mathematical model for the quadrotor is developed, its parameters determined from the characterization of the unit. An external intelligence is integrated to play the rol 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. For windless environments, a stable platform is achieved.**

I. INTRODUCTION

A quadrotor platform can be imagined performing countless tasks. 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 [4] [5] to represent the system, the filtering techniques applied for sensor data fusion [6] [7] and the control system [4] [8] that allows the quadrotor to fly.

The platform is based on a commercial radio controlled quadrotor, the frame and the motor control system were preserved, 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 [9] [10] is used to derive the control actions required to bring the system to the desired setpoint.

II. MODEL OF A QUADROTOR

A. Definitions

A diagram of the quadrotor is shown in figure (1). Two of the motors rotate clockwise (2 and 4) and the other two (1 and 3) rotate clockwise. This configuration allows the quadrotor to rotate, tilt and gain/lose altitud by setting different speeds on each motor. Two frames of reference (figure 1) are constantly used through out this paper: An intertial frame S_I, relative to the Earth, and a non-intertial frame S_q, relative to the quadrotor. The mapping of one frame to the other can be achieved by applying the three rotations shown in figure (2). The angles $\{\theta, \varphi, \psi\}$ are known as Euler angles.

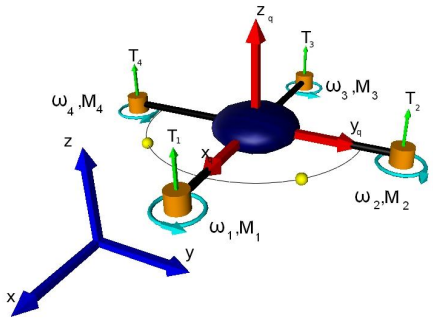
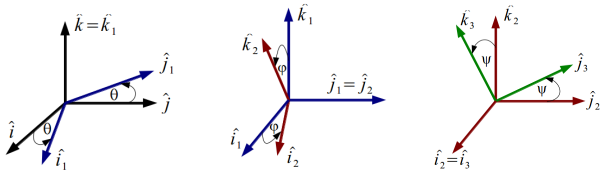


Fig. 1: **Model of the quadrotor -** The blue arrows represent the intertial reference frame $S_I - \{\hat{i}, \hat{j}, \hat{k}\}$ ($\{\vec{x}, \vec{y}, \vec{z}\}$), mapped to North, West and Up respectively. The red arrows represent the non-inertial reference frame $S_q - \{\hat{i}_q, \hat{j}_q, \hat{k}_q\}$ ($\{\vec{x}_q, \vec{y}_q, \vec{z}_q\}$) relative to the quadrotor. The cyan "looped" arrows indicate the direction of rotation of each motor. The arrows labeled $T_{[1,2,3,4]}$ represent the thrust of the motors. The semicircle and the two yellow spheres indicate the front of the unit.



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

Fig. 2: **Mapping:** Rotations applied on S_I to obtain S_q.

B. Dynamics-kinamatics 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

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

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:

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

where:

- $\{x, y, z\}$ are the cartesian coordinates in S_I.
- $\{\theta, \varphi, \psi\}$ are the Euler angles show in ((2)).
- $\{v_{q_x}, v_{q_y}, v_{q_z}\}$ are the linear velocities relative to S_I.
- $\{w_{q_x}, w_{q_y}, w_{q_z}\}$ are the angular velocities relative to S_q (right hand rule applied on $\{\hat{i}_q, \hat{j}_q, \hat{k}_q\}$).

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

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 $\{v_{q_x}, v_{q_y}, v_{q_z}\}$, 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, which is used internally by the barometer. 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. A calibration [1] 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, it is used to determine two of the three Euler angles: $\{\psi, \phi\}$. This hypothesis is acceptalbe, since the rest of the accelerations involved are not significant compared to gravity. A calibration [1] 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 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 was performed on this sensor, based on [2] and [3].

B. GPS

A GPS is used to determine the absolute position of the system, since the IMU cannot provide this information. The accuracy under 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.

IV. Kalman Filter

In order to perform adequate control actions, a reliable estimation of the state variables must be performed, 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

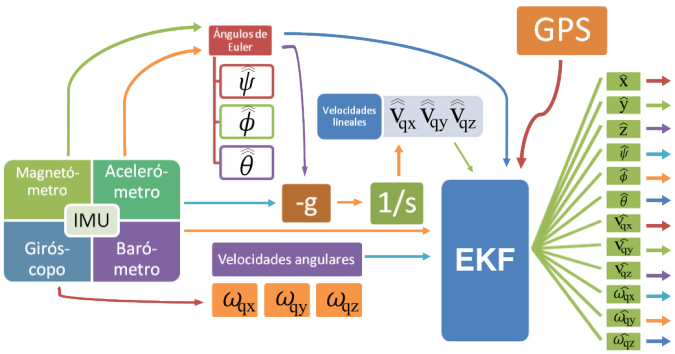


Fig. 3: **EKF -** Outline of how sensor data is combined to estimate the state variables.

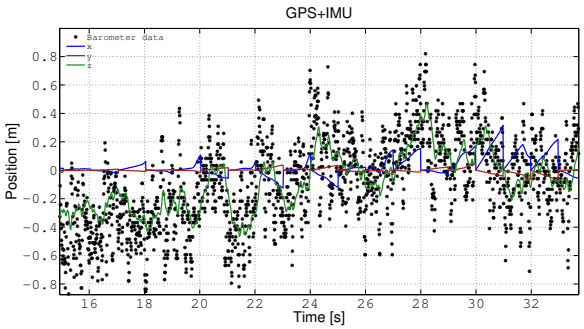


Fig. 4

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 very little precision 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.

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 [6] and [7]. 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.

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

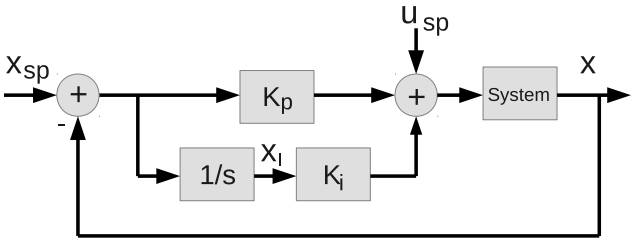


Fig. 5: **Control System:** x_{sp} and u_{sp} represent the setpoints for the state vector and the speed of each motor. The K_p and K_i blocks are the proportional and integral gain matrices. The output of the system is the current state vector.

integration, so only the prediction step can be performed for these variables. As shown in figure (4), the estimation will drift until GPS data is available.

Figure (3) 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. This restricts the system to: Hovering[4], uniform linear trajectories, and uniform circular trajectories. This choice 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

For a system which is controllable and observable [8], the LQR algorithm finds an optimal controller [9] by minimizing the following cost function:

$$\int_0^\infty \vec{X}'(t)Q\vec{X}(t) + \vec{U}'(t)R\vec{U}(t)dt \qquad (3)$$

[4]Constant state vector.

where Q and R are positive-definite matrices, representing the energy of the controlled states and 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 an error.

When the system is a feedback system with known state variables, like the one shown in figure (5), the solution is given by (4), where P is the solution to Ricatti's equation.

$$\begin{aligned} \vec{U}(t) - \vec{U}^*(t) &= -K(\vec{X}(t) - \vec{X}^*(t)) \\ K &= R^{-1}B^T P \end{aligned} \tag{4}$$

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

The software runs on several independant boards: An ARM-Cortex-A8 on the flight controller, an ATMega328p on the IMU, and four microprocessors[5] on the electronic speed controller (ESC), one for each motor. The code running on the ESCs is not available, but the communication protocol was decoded using a logic analyzer. 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. The code running on the flight controller 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.

VII. Flight tests

Many flight tests were performed, most of them with strings attached above and below the quadrotor as security measures. The poor reliabitily of the ESCs provided by LotusRC[6] made testing a dangerous.

respuesta al escalon de las cosas?

resultados buenos para lugares sin viento, posibles explicaciones

limitaciones por culpa de los ESCs

agregar ejemplos de circulos, ver q la altura

A. Subsection Heading Here

Subsection text here.

1) Subsubsection Heading Here: Subsubsection text here.

VIII. Conclusion

The conclusion goes here.

A. Future Work

- The LotusRC ESCs proved to be extremely unreliable[7]. Random glitches cause the motors to turn off, which inevitably leads to an accident. 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. Implementing the stabilization system, which requires a very fast and steady loop, on a dedicated microprocessor would have been a better approach. This 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 control system does not work in a windy environment severly limits the range of possible applications. Modifying the control system to allow flight under windy conditions would be a significant improvement.

Acknowledgment

The authors would like to thank de panader.

References

[1] I. Skog and P. Hndel, "Calibration of a mems inertial measurment unit," in *XVII Imeko World Congress, Metrology for a Sustainable Development*, (Rio de Janeiro), Setiembre 2006.

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

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

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

[5] M. Vendittelli, "Quadrotor modeling." Curso: "Elective in robotics", Sapienza Universit Di Roma, Noviembre 2011.

[6] 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, Mayo 2012.

[7] 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, Mayo 2008.

[8] M. Hakas, "Sistemas de control en tiempo discreto." Curso: "Introducción a la Teoría de Control", Facultad de Ingeniería Universidad de la República, 2005.

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

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

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

[5] C8051F330/1 - Silicon Labs.

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

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