Design and implementation of flight control for an autonomous quadrotor

Santiago Paternain, Matías Tailanián, Rodrigo Rosa, Rafael Canetti

***Abstract*— In this paper we describe the stages needed to achieve an autonomous stable platform with quadrotor architecture. Based on a commercial frame, we first design and integrate the instrumentation. An Inertial Measurement Unit (IMU), which also has a barometer, is added to the platform and accurately calibrated. The sensor fusion and signal filtering are done based on an Extended Kalman Filter (EKF). Finally, a proportional-integral controller based on an enhanced Linear Quadratic Regulator (LQR) algorithm is implemented in the control system, using the mathematical model of the quadrotor. The parameters of this model are obtained from the characterization of the unit, achieving at the end, a stable autonomous platform.**

I. Introduction

In the past years, there has been growing interest in unmanned aerial vehicles (UAV). They can be used in an extensive range of applications and fields; from the most basic ones like aerial photography, monitoring or surveillance, to more sophisticated ones, like remote sensing, exploration, delivery, or even rescue. The continuous technological advances that are currently made in low power MEMS sensor devices and embedded processors, energy storage and efficient electric batteries and motors, are powering the implementation of even more advanced UAVs, with special emphasis in miniature flying robots. Quadrotor UAV is one of the most popular of these architectures, having a great proliferation in the past years. They are also used as research platforms to develop computer vision techniques, path planning algorithms towards navigation in unstructured scenarios [1]–[3], path tracking [4], fault tolerant navigation [5], control techniques for UAV such as neural networks [6], gain-scheduling [5], feedback linearization [3], [7], PI control [4], Adaptive Control [8], LQR [2], back-stepping control [1], [3], etc. The goal of this work is to design, develop and implement the instrumentation, the sensor fusion and filtering algorithms, and the control system of an autonomous quadrotor. Each part of the system is done modularly, to ease the substitution of each component, allowing to test different path planning and control techniques in a real physical setup, while keeping the access to the different blocks of the navigational system. The solution is based on a commercial platform, adding the ad-hoc designed instrumentation and control navigation system. We describe the system in Section II. The focus of this paper is in the stabilization of the quadrotor (Section V), as the aerial vehicle is inherently unstable. The mathematical model developed for the system representation (Section III) and the filtering techniques applied to fuse the sensors and obtain a more reliable measure (Section IV) are key components required for the control of the platform.

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

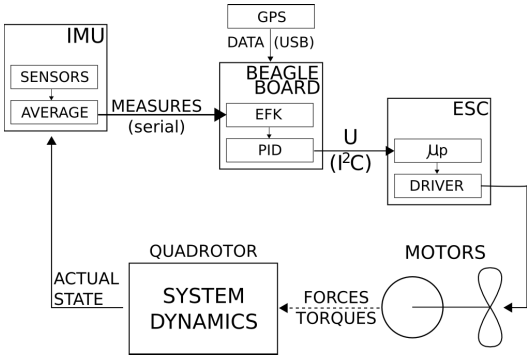


Fig. 2: **General diagram.**

II. System architecture

Our solution is based on the commercial radio-controlled quadrotor shown in Figure (1). It has a payload of $1300g$ and it weights $990g$, including the battery. The distance between opposite propellers is $61.5cm$. From the original assembly, only the frame, motors and Electronic Speed Controllers (ESCs) were kept. The IMU and intelligence were replaced with the flight controller that was developed. All the computations are made in a BeagleBoard[1] running Linux[2], from the conversion of the raw data received from the IMU[3] over a UART, the filtering and sensor fusion made with the EKF, to the control algorithm of the system. The EKF combines the sensors data in a way that takes the advantage of each sensor, overcoming the problem inherent to each sensor, filtering noise and providing a reliable estimation of the state vector, which the LQR algorithm uses to derive the control actions required to take the system to the desired set-point. In (2) is showed a complete diagram of the system implemented.

[1]BeagleBoard development board - `http://beagleboard.org/`

[2]Angstrom distribution: `http://www.angstrom-distribution.org/`

[3]*Mongoose* IMU - `http://store.ckdevices.com/`

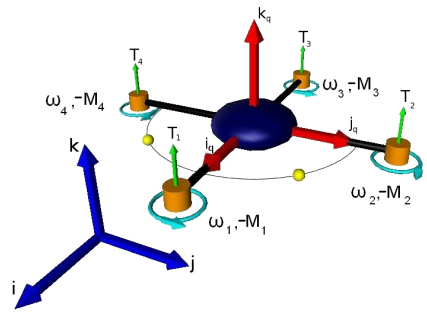
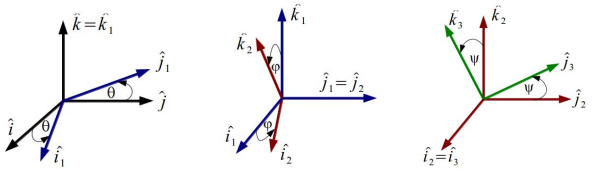


Fig. 3: **Model of the quadrotor -** The cyan arrows indicate the direction in which each motor rotates. By rotating at the angular speed ω_i they generate a torque M_i opposite to their direction of rotation and a thrust T_i perpendicular to the plane defined by the propellers. The thrust is represented by the arrows labeled $T_{[1,2,3,4]}$. The direction from the center of the quadrotor to the propeller in the center of the semicircle and the yellow spheres is the direction $\hat{i}_q$



(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.

III. Model of a Quadrotor

Each motor of the quadrotor generates a thrust perpendicular to the plane of the blade of the propellers and directed upwards as shown in Figure (3). The thrust of each propeller can be modeled as a third order polynomial of its angular speed and thus, by controlling the speed it is possible to unbalance the thrust of each propeller which allows the system to tilt and modify its altitude. In addition, the motors 2 and 4 rotate clockwise while the motors 1 and 3 rotate counterclockwise as it can be observed in Figure(3). This produces a torque in the direction perpendicular to the plane of the propellers that depends on the difference of angular speed between each pair of motors, which allows the quadrotor to rotate along the vertical axis. To be more precise, let us define the an inertial frame of reference $S_I - \{\hat{i}, \hat{j}, \hat{k}\}$, relative to the Earth, mapped to North, West and Up respectively, and a non-inertial frame $S_q - \{\hat{i}_q, \hat{j}_q, \hat{k}_q\}$ relative to the quadrotor. To express any quantity in or the other frame of reference it suffices to perform three rotations as described in Figure 4.

We denote by $\{x, y, z\}$ the position of the center of mass of the quadrotor expressed in the inertial frame and by $\{\theta, \varphi, \psi\}$, the Euler angles defined by the rotations in Figure 4. Finally, the linear and angular velocities expressed in the frame of the quadrotor S_q are denoted respectively by $\{v_{q_x}, v_{q_y}, v_{q_z}\}$ and $\{w_{q_x}, w_{q_y}, w_{q_z}\}$. With these definitions, one can represent the variables defining the state space of the system in a compact form by choosing $\mathbf{x}$ to be

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

In addition, we denote the thrust and drag of each motor by T_i and Q_i with $i = 1 \ldots 4$. The kinematics and the dynamics of the quadrotor satisfy the following system of dynamical equations

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

The previous model is similar to those in the literature [9], [10]. The main difference is that in our model the center of gravity of the quadrotor is not assumed to lie in the plane defined by the propeller. The latter entails an additional torque produced by the gravity that affects the dynamics of ω_{q_x} and ω_{q_y}. A second difference is that we express the velocities in the body frame whereas in [10], these are relative to the inertial frame. The reason for our choice is that the linear accelerations in the body frame can be directly measured by the accelerometers in the Inertial Measurement Unit mounted on the quadrotor.

IV. Sensors Fusion

A fundamental component in control systems is the adequate estimation of the state variables of the system. Regardless how well designed a given control law is, without accurate state estimation the system might be unstable. In addition, if the estate is estimated with great precision but the estimates are available with large delay, no stability guarantees can be provided. In that sense, having a real-time and accurate estimation of the state variables is a

cornerstone of every control system. In this work, we rely on an extended Kalman Filter (EKF) implemented as described in [11]. The two main components of a Kalman Filter are the prediction and the correction. In the former, given the current estimation of the state, one predicts, based on the dynamics of the system, the state at the next time. In the latter, using the measurements of the sensors, the prediction is corrected. Each sensor provides measurements about different variables. In particular, a 3-axis accelerometer allows to estimate the Pitch and the Roll (ψ and ϕ), and combined with a magnetometer one can measure the Yaw (θ). The details of such estimation are provided in [11]. The 3-axis gyroscope allows measurement of the angular speeds, the barometer is used to estimate the height based on changes of air pressure and the GPS is used to determine the absolute position of the center of mass of the system x and y. The calibration of these sensors –for which we provide a summary of features in Table I – is done as in [12].

	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:** "*10ms (x2)*" means that every 10ms the IMU outputs the average of 2 samples.

V. Control design

We derive a linearized model of the quadrotor dynamics (2) and we restrict our attention to three types of trajectories: hovering, uniform linear trajectories, and uniform circular trajectories around the vertical axis. The restriction to these trajectories yields a Linear Time Invariant system of the form

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}. \tag{3}$$

The control system is shown in Figure (5). The design of the feedback matrices K_p and K_i is done using a Linear Quadratic Regulator (see e.g. : [13]). In particular, we consider an extended state space vector so to include the integrals of some state variables as states, which allows for integral control action. The introduction of such variables mitigates error in the characterization and modeling of the system.

A system which is controllable and observable is guaranteed to be stable when controlled by an LQR (see e.g. [13]). Let $\mathbf{x}_{sp}$ be the objective state and $\mathbf{u}_{sp}$ the control action that is compatible with such state and let $\mathbf{Q}$ and $\mathbf{R}$ are positive-definite matrices, weighing 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. The LQR controller is such that

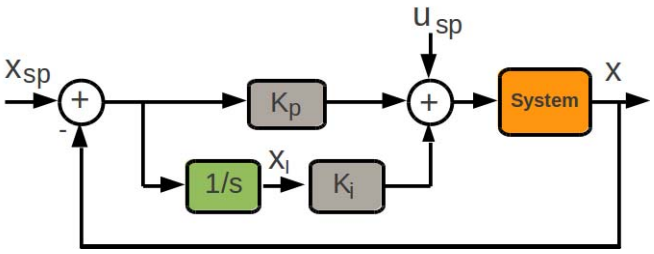


Fig. 5: **Control System:** X_{sp} and U_{sp} represent the set-points for the state vector and the speed of each motor. The K_p and K_i blocks are the proportional and integral gain matrices. The output of the system is the current state vector.

it minimizes the following performance cost

$$\int_0^\infty (\mathbf{x}(t)-\mathbf{x}_{sp})^T\mathbf{Q}(\mathbf{x}(t)-\mathbf{x}_{sp})+(\mathbf{u}(t)-\mathbf{u}_{sp})^T\mathbf{R}(\mathbf{u}(t)-\mathbf{u}_{sp})\,dt, \tag{4}$$

by selecting the optimal control input

$$\mathbf{u}(t) = \mathbf{u}_{sp} + \mathbf{K}(\mathbf{x}(t) - \mathbf{x}_{sp}), \quad \text{with} \quad \mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}. \tag{5}$$

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

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = 0. \tag{6}$$

A. Integral effect

In this work, an integral effect is added to 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 used for feedback in the integral control block, these are $\dot{\mathbf{x}}_I = [x, y, z, \theta]^T$. With this definitions, we follow the same ideas as in (4)–(6) where now the state has been enlarged and the matrix $\mathbf{K}$ is composed now of both $\mathbf{K}_p$ $\mathbf{K}_i$.

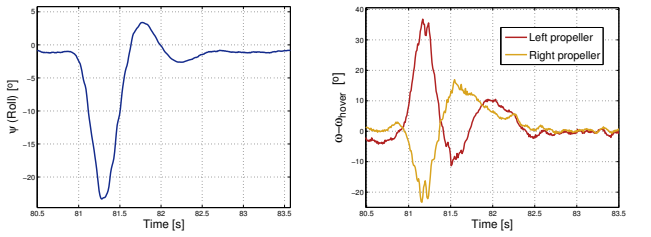
B. Implementation details

The code implements a discrete version of the LQR control system mentioned in (V), based on a discretization of the model described in (III). Two practical considerations are worth mentioning. In order to avoid divergence of the integral terms, an Anti-wind-up algorithm was implemented and a limiting in the control actions, where we establish a threshold $|\vec{X} - \vec{X}_{sp}| < |\vec{X}_{th}|$ to avoid attempting to perform actions that could entail large accelerations.

VI. Software

The software runs on several independent boards:

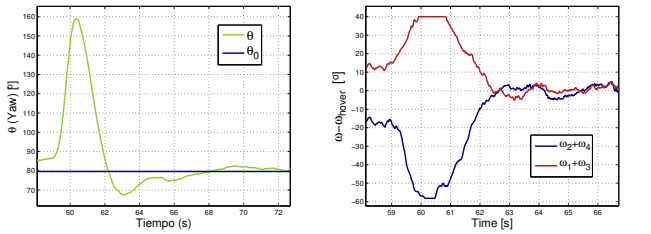
- **Flight controller:** Run by an ARM-Cortex-A8. Processes the data from the IMU calculates the necessary control actions and sets the desired speed for each motor by sending the appropriate commands, via I^2C, to the ESCs. It requires Linux specific C functions to handle I/O. The design of the flight 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.



(a) Roll angle deviation

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

Fig. 6: **Stabilization -** The Roll angle is deviated from equilibrium and the quadrotor manages to stabilize itself.



(a) Yaw angle deviation

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

Fig. 7: **Orientation -** The Yaw angle is deviated from equilibrium and the quadrotor manages to stabilize itself.

- **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[4], one for each motor.

One can log-into the flight controller via an *ssh* session over *WiFi* where the software can be compiled and executed.

VII. Experiments

A. Basic stability

We focus first on the control of the angles $\{\psi, \phi\}$. The quadrotor was allowed only to move along one of those angles at a time to tune the controller. Once this stage was completed the quadrotor was able to keep a horizontal position by controlling those two Euler angles and the angular velocities $\{\omega_{q_x}, \omega_{q_y}\}$. The response of the control system to a perturbation for the Roll is shown in Figure (6). In Figure (6a) it is possible to observe the mechanical perturbation imposed from $t_0 = 80.9s$ to $t_1 = 81.3s$ and how the system stabilizes after releasing the system, while in Figure (6b) the control inputs required to stabilize the system are shown. The response of the quadrotor for a step of more than 20^o shows an overshoot of 3^o or 4^o and a rise time of less $0.4\ s$.

B. Orientation and height control

Figure (7) shows the system behavior against a mechanical perturbation (from $t_0 = 59.3s$ to $t_1 = 60.5s$) in the 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 $1.6\ s$, higher than in the other two Euler angles, which can be allowed as Yaw angle is not as critical for stability. Likewise, the overshoot is about 13% of the step. Once attitude control is shown to perform adequately we proceed to test the quadrotor's flight. As observed in Figure (8) the system is stabilized at the goal height of one meter.

[4]C8051F330/1 - Silicon Labs.

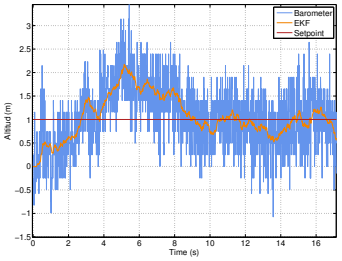


Fig. 8: **Altitude:** Performance during takeoff.

VIII. Conclusion

A controller based on a linear quadratic regulator with integral control action was designed for a quadrotor. Attitude and height tests show good performance of the system.

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

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

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

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

[11] M. Tailanian, S. Paternain, R. Rosa, and R. Canetti, "Design and implementation of sensor data fusion for an autonomous quadrotor," in *Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, 2014 IEEE International*, pp. 1431–1436, IEEE, 2014.

[12] S. Paternain, M. Tailanian, and R. Canetti, "Calibration of an inertial measurement unit," in *Advanced Robotics (ICAR), 2013 16th International Conference on*, pp. 1–6, IEEE, 2013.

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