Study and Experimentation of Control Policies to Dynamically Maintain Micro-UAV Flight Stability

Federico Pedersini*
Università degli Studi di Milano
pedersini@di.unimi.it

Andrea Toscano
Università degli Studi di Milano
andrea.toscano89@gmail.com

Elena Pagani
Università degli Studi di Milano
pagani@di.unimi.it

1. INTRODUCTION

Unmanned Aerial Vehicles (UAVs or drones, for short) are receiving increasing attention recently, due to the many applications they might be used for, ranging from territory surveillance to delivery of goods. The case of drones autonomously flying according to a pre-defined route is clearly the most interesting one for the deployment of such applications. Yet, autonomic flight requires that a drone is able to maintain stability – in terms of a target attitude – in spite of external disturbances (e.g. gusts of wind), *with no human intervention*. This is traditionally done with a *PID* (proportional - integral - derivative) *controller*, which takes as input the deviations from the target, and supplies in output the indication of how to act on the drone engines so as to restore the proper attitude. Two orders of problems must be dealt with when deploying a PID controller. First, the parameters weighing the input components must be properly tuned in order to guarantee attitude restoring, and to avoid oscillations of the system. Second, the PID output must be translated into commands to the drone engines so as to achieve the desired behavior. The former aspect can be solved either by hand, or with a number of automatic methods proposed in the literature (e.g. [1, 2]). Those methods involve complex mathematical models and are considerably time-consuming. As far as the latter aspect is concerned, there are hardware constraints that impede to abruptly change speed or spin of rotors and propellers, which otherwise may be damaged.

In this paper, we introduce a mathematical model simplified yet effective in yielding appropriate parameters to implement an accurate controller, without the need of a complex preliminary calibration of the mechanical system. We describe our policies to apply controller indications to the drone hardware. We validate both model and policies through experiments.

2. THE PROPOSED SYSTEM

In a quadcopter, each rotor generates a force $\mathbf{F}$, parallel to the rotation axis, and a torque $\mathbf{T}$. In conditions of "static flight", called *hovering*, the resultant of the four forces equilibrates the gravity and the four torques sum up to a null momentum, as the rotors run alternatively clockwise and counter-clockwise. A horizontal motion parallel to the x axis is obtained by tilting the quadcopter around the y axis, i.e. setting the *pitch* angle to a non-zero value. The four rotor forces are then tilted towards x, so that a horizontal component appears, causing acceleration towards x, while the vertical component still equilibrates gravity. In the same way, a non-zero *roll* angle causes a horizontal acceleration along y. Vertical motion is obtained by changing the intensity of all four forces by the same amount $\Delta\mathbf{F}$. Assuming an initial hovering state where $\sum \mathbf{F}_i = mg$, if $\Delta\mathbf{F} > 0$, the resulting force accelerates the vehicle upwards, for $\Delta\mathbf{F} < 0$ downwards. The intensities of $\mathbf{F}$ and $\mathbf{T}$ generated by each rotor are a function of the rotation speed; therefore the quadcopter motion can be regulated by controlling the speed of its four rotors. The relationship between thrust and speed is rather complex and depends on many factors. A good approximation is [1]: $F \propto \eta(\omega) \cdot \omega^2$, where ω is the angular rotor speed and η is the propeller efficiency, which itself depends on ω. However, since the proposed control system works by applying small adjustments ΔF to the rotor forces, we adopted a linear model for this relationship, that is, simply: $\Delta F = k_{pr} \cdot \Delta\omega$. Under these assumptions, the motion of the vehicle can be controlled by properly changing the rotor speeds, according to the scheme summarised in Table 1. A big advantage of having a linear relationship is that

Table 1: speed controls for motion

	rotor 1	rotor 2	rotor 3	rotor 4
roll	$+\Delta\omega_r$	–	$-\Delta\omega_r$	–
pitch	–	$+\Delta\omega_p$	–	$-\Delta\omega_p$
yaw	$-\Delta\omega_y$	$+\Delta\omega_y$	$-\Delta\omega_y$	$+\Delta\omega_y$
thrust	$+\Delta\omega_t$	$+\Delta\omega_t$	$+\Delta\omega_t$	$+\Delta\omega_t$

the motion control can be carried out separately for each kind of motion (roll, pitch, yaw, thrust) and then the resulting actions can be simply linearly superimposed. This choice greatly simplifies the control system, which can be decomposed into four separate one-dimensional (and therefore much simpler) controls.

2.1 The control system

The functional diagram of the proposed control system is presented in Fig. 1. The system is composed of four independent channels, each controlling one parameter: roll, pitch, yaw, and thrust. The outputs of each channel are then linearly composed to give the four desired rotor speeds $\hat{\omega}_i$. The heart of each control channel is a PID; at each time

DroNet'15, May 22, 2015, Florence, Italy.
ACM 978-1-4503-3501-0/15/05.
http://dx.doi.org/10.1145/2750675.2750676.

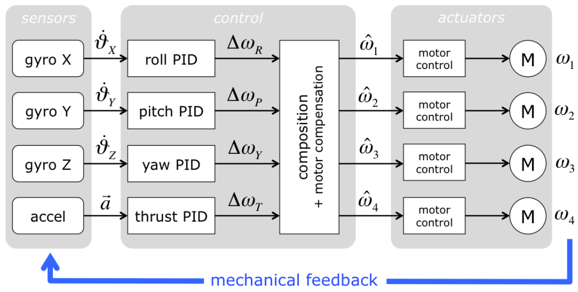

Figure 1: Schematics of the control system.

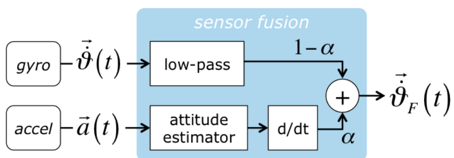

Figure 2: Sensor fusion architecture.

instance, it generates the current value for the requested speed change, $\Delta\omega_{r/p/y/t}$, based on its input. The four inputs are, respectively, the three components of the vehicle's angular velocity, $\dot{\vartheta}_X$, $\dot{\vartheta}_Y$, $\dot{\vartheta}_Z$, and the acceleration vector $\vec{a}$. By mounting the quadcopter onto particular setups that constrain the vehicle to move on one axis only (x for roll, y for pitch, z for yaw), each PID can be tuned independently from the others. For the PID tuning, we adopted a heuristic approach, which allows to avoid complex mechanical characterisations of the vehicle. We followed the Ziegler-Nicholls method [2], obtaining satisfying results, as described in the next section. As the diagram shows, the actuators (motor controllers and rotors) are a part of the dynamical system. However, their behaviour cannot be modified and is actually unknown. In order to identify this part of the system, a dynamical characterisation of the rotor would be necessary, which would be complex and expensive. Actually, due to electro-mechanical constraints, the motor system behaves dynamically as a 1-pole system with a cut frequency around 1 Hz (depending on motor, propeller, and motor controller); this causes a significant "smoothing" effect on the resulting speeds $\omega_i(t)$, with respect to the speeds requested by the control system, $\hat{\omega}_i(t)$; this greatly reduces the margins of stability for the control system. This effect has been successfully reduced by applying to the output signals $\hat{\omega}_i(t)$ a filtering to partly cancel the motor pole.

2.2 Sensor fusion

The quality of the attitude signal measured by the sensors is of crucial importance, as it has a direct influence on flight stability. Normally, the quality of the raw measurements produced by commercially available IMU's is not sufficient, due to superimposed noise, sensor offsets and drifts. For this reason, in order to get sufficient accuracy from low-cost sensors, we adopted a sensor fusion approach. As shown in Fig. 2, a better estimate of the current angular rate is obtained as weighted sum of the gyroscope output after a proper low-pass filtering (to eliminate the spurious signal due to vibration) and a derived estimate of the angular rate, obtained from the accelerometer by estimating the current vehicle attitude as direction of the gravity vector and then computing its time derivative around the three axes. Sensor fusion is widely adopted in UAV control. It needs accurate tuning for the weights and the filters, but can provide good measurements from low-cost sensors.

3. EXPERIMENTAL RESULTS

In this section, we discuss the experiments we conducted to validate our simplified model. We built a custom quadcopter using off-the-shell components. As the microcontroller, we used an Arduino Yun board. The microcontroller takes as input the signals sent by a GY80 IMU (Inertial Measurement Unit), including a 3-axis angular rate sensor and a 3-axis digital accelerometer. No GPS device is used. As preliminary steps, we implemented the simplified model in the control system and we submitted the UAV to solicitations simulating external disturbances. In order to test the capability of maintaining flight stability, the drone was fixed either to a rotating platform, which was submitted to displacements around its vertical axis, or to a long stick suspended between two supports so as to be able to rotate around its own axis. In both cases, in the absence of the filter, the drone comes back to the target position with small errors. Occasionally, minor oscillations were observed before halting in an approximately correct position. By contrast, the use of the filter resulted in both greater inertia in front of external impulses, and lower hysteresis in correcting the attitude: the drone resisted to the rotations with prompt adaptation of the rotors thrust, a greater effort of the experimenter was needed to displace the drone, and return to the target position was faster and more precise with no oscillations. Symptoms of integer wind-up and instability were never observed. Once gained confidence in the control system and parameter tuning, we performed some experiments outdoor. The drone was led to an altitude of around 30-40 m., in order not to loose the WiFi connection with it, and there "parked" with no human intervention in order to verify the capability of the control system to autonomously maintain flight stability in spite of wind. According to the local weather service, in the days of outdoor experiments wind was ranging from calm to light breeze. The observations showed that the drone was very stable, with an ability of maintaining the position comparable to that of a much more expensive (and likely accurately tuned) drone available in commerce equipped with a GPS device, thus validating our simplified model for a PID controller. Some videos of the experiments described in this work are available in YouTube, under the label "Comelicottero".

4. CONCLUSIONS

In this paper, a simplified mathematical model to implement and tune a PID controller for a quadcopter is presented. The model is validated through experiments conducted both indoor and outdoor, which always showed that the output produced by the controller is able to maintain flight stability without any human intervention.

5. REFERENCES

[1] A. Gibiansky, *Quadcopter Dynamics, Simulation, and Control*, http://andrew.gibiansky.com/blog/physics/quadcopter-dynamics

[2] J.G. Ziegler, N.B. Nichols, Optimum settings for automatic controllers, *Trans. ASME 64*, pp. 759–768, 1942.