EPiC Series in Computer Science
Volume 34, 2015, Pages 57–72
ARCH14-15. 1st and 2nd International Workshop on
Applied veRification for Continuous and Hybrid Systems

Benchmark: Quadrotor Attitude Control

Antonio Eduardo Carrilho da Cunha[1]*

Instituto Militar de Engenharia (IME)
Praça General Tiburcio, 80, Praia Vermelha
22.240-260, Rio de Janeiro, RJ, Brasil
carrilho@ime.eb.br

Abstract

We present a case study of attitude control for a quadrotor drone and propose the application of reachability analysis to investigate and improve the robustness of the control design. The controller is to be implemented and tested using an experimental platform, the *CrazyFlie*. We intend to use measured data to improve the models employed in reachability analysis.
Category: academic **Difficulty:** medium

1 Context and Origins

Recent advances in low-power embedded processors, wireless communications and miniature sensors and actuators have increased the interest on the development of drones with a wide range of indoors and outdoors applications: safety, security, defense, inspection, communication links, data acquisition, entertainment, package delivery, not being exhaustive.

Quadrotors are of particular interest when implementing small-scale drones because of the simpler control and stabilization mechanisms and their ability to perform vertical take-off and landing (VTOL), omnidirectional movements, hovering, or low speeds flights [1]. One disadvantage is the high power consumption during the flight.

When approaching the control of quadrotors, we can identify two main problems: the *attitude stabilization* and the *guidance* [2]. The attitude stabilization aims to enhance the vehicle dynamics by feedback control. It can be employed to enhance stability in remote piloting. The guidance has a broader objective that is to control the position and the orientation of the vehicle, aiming the autonomous behavior. Usual approaches in guidance are *interception*, *surveillance* or *rendez-vous* [2]. In general, the attitude stabilization and the guidance problems are treated by separated control loops, being an inner loop related to the first, and an outer loop to the second. Generally, the time constants of the two problems are different, being the attitude stabilization with faster dynamics than the guidance.

We develop a case study of attitude control of a quadrotor. In our approach we propose a nested control architecture exploiting the modularity of the model. An inner loop takes care of the angular rates and vertical speed, and an outer loop takes care of the angular orientation

*Work developed as a visitor researcher to VERIMAG, France.

and position. In order to investigate the robustness of the controller, we apply reachability analysis [3,4]. We present two nonlinear models for the quadrotor, one based on the Euler angles and the other on quaternions. A linearized model obtained from the Euler angles model is used for control design. On the other hand, a linearized model obtained from the quaternions model is used for reachability analysis. The quaternions model is polynomial and singularity free, when compared to the Euler angles model. This enables the application of many reachability analysis techniques suitable to polynomial systems as, for example, [5] or [6]. In this work we overapproximate the polynomial nonlinearities using McCormick relaxations [7]. Moreover, we will implement the controller into a real experimental quadrotor platform, the *CrazyFlie*, from *BitCraze*[1], and use measured data to improve the model employed in the reachability analysis. Identification techniques for piecewise affine systems as in [8] are to be used in this step.

Another benchmark for reachability analysis based on a quadrotor is presented in [9]. They define a flight envelope protection used for safety-preserving controller synthesis. In their definition of the flight envelope, they address both attitude stabilization and guidance problems. But they present a nonlinear model that disregards many of the relevant dynamic interactions for the attitude stabilization problem, for example, the Coriolis forces in translational and rotational dynamics. Therefore we believe that the model in [9] is more suited to address guidance problems, while the model presented in this paper is suitable for both attitude stabilization and guidance problems.

This paper is organized as follows. Section 2 presents the quadrotor model and the attitude controller design. Section 3 introduces our approach for applying the reachability analysis. Section 4 presents some remarks on the current results of the work.

2 Brief Description

2.1 Quadrotor Nonlinear Model

Figure 1 shows a top view of a quadrotor. Consider a quadrotor where each rotor R_i, $i \in \{1, 2, 3, 4\}$, produces on the airframe a force F_i and a torque τ_i, according to its direction of rotation. The front and back rotors spin clockwise and the right and left rotors spin counterclockwise [1]. We treat a quadrotor with a *cross* "+" configuration, so that the *front* of the quadrotor points towards rotor R_1, the *back* to rotor R_3 and, when looking from above, the *right* to rotor R_2 and the *left* to rotor R_4.

For control purposes, the model inputs are the forces and torques produced by the rotors on the airframe [1]: the *Thrust* $F = F_1 + F_2 + F_3 + F_4$, the *Roll Torque* $\tau_\phi = l(F_4 - F_2)$, the *Pitch Torque* $\tau_\theta = l(F_3 - F_1)$, and the *Yaw Torque*: $\tau_\psi = \tau_1 - \tau_2 + \tau_3 - \tau_4$, where l is the distance from the rotors to the quadrotor center of mass.

It is important to relate the forces applied to the quadrotor by the rotors to the actual command sent by the controller board. In general, the controller board sends a step δ_i, to a Electronic Speed Controller (ESC) that, in turn, applies a voltage v_i to the rotor R_i. This makes the propeller to rotate in an angular speed ω_i and, therefore produce an aerodynamic force F_i and torque τ_i in the frame. The relations $\delta_i \times F_i$ and $\delta_i \times \tau_i$ are fundamental to make the controller give the correct commands to the rotors and should be experimentally obtained [1,10].

There are three relevant reference frames for a quadrotor model. The Earth is supposed to be flat, as a consequence of the size of the device and the length of the missions. In the *Inertial Frame*, $\mathcal{F}_i$, also referred to as NED Frame, the origin is fixed on a reference point on

[1] http://www.bitcraze.se/

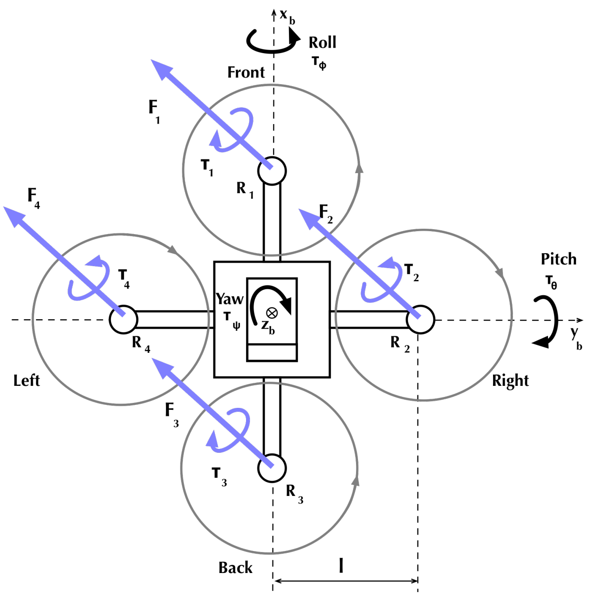


Figure 1: Quadrotor "+" configuration.

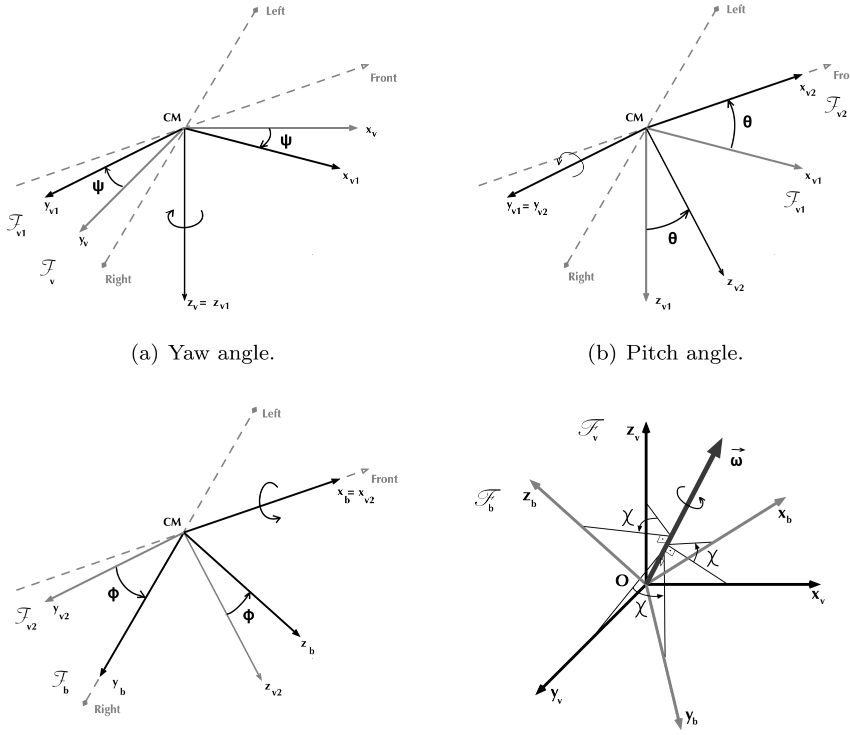
the ground; axis x_i is aligned to the North direction; y_i is aligned to the East direction; and z_i is aligned to the vertical downard direction. The *Vehicle Frame*, $\mathcal{F}_v$, is parallel to the inertial frame with origin in the center of mass of the airframe. Finally, the *Body Frame*, $\mathcal{F}_b$, Figure 1, has origin in the center of mass of the airframe, axis x_b points towards the airframe front; y_b points to the right of the airframe; and z_b points downwards.

The state variables for the quadrotor model are the following:

- The *inertial position* $\vec{P} = [p_n \ p_e \ h]^T$ is measured with respect to the inertial frame $\mathcal{F}_i$, where p_n is the north position; p_e is the east position; and h is the height ($-z_i$ direction).[2]
- The *linear velocity* $\vec{V} = [u \ v \ w]^T$, is measured with respect to the body frame $\mathcal{F}_b$, where u is the x_b-component; v is the y_b-component; and w is the z_b-component.
- The *angular orientation* is represented by quaternions and Euler angles. The *quaternion* $Q_b^v = [q_0 \, q_1 \, q_2 \, q_3]^T$ represents the orientation of the body frame with respect to the inertial frame. The *Euler angles* are $\Sigma = [\phi \ \theta \ \psi]^T$, where ϕ is the roll angle, θ is the pitch angle and ψ is the yaw angle.
- The *angular velocity* $\vec{\Omega} = [p \ q \ r]^T$ is measured with respect to $\mathcal{F}_b$, where p is the x_b-component, or *roll* rate; q is the y_b-component, or *pitch* rate,; and r is the z_b-component, or *yaw* rate.

The *Euler angles* $\Sigma = [\phi \ \theta \ \psi]^T$ are defined from a sequence of rotations on the vehicle frame $\mathcal{F}_v$ in order to obtain the body frame $\mathcal{F}_b$, Figures 2 (a), (b) and (c). First, frame $\mathcal{F}_v$ is right-hand rotated around axis z_v by the *yaw angle* ψ, obtaining frame $\mathcal{F}_{v1}$ by making axis x_{v1} point towards the projection of the airframe *front* in the $x_v \times y_v$ plane , Figure 2(a). Second, frame $\mathcal{F}_{v1}$ is right-hand rotated around axis y_{v1} by the *pitch angle* θ, obtaining frame $\mathcal{F}_{v2}$ by

[2] Details on the modeling are presented in the accompanying expanded version of the paper.



(a) Yaw angle.

(b) Pitch angle.

(c) Roll angle.

(d) General rotation.

Figure 2: Euler Angles and Quaternion.

making axis x_{v2} point towards the airframe *front* , Figure 2(b). Finally, frame $\mathcal{F}_{v2}$ is right-hand rotated around axis x_{v2} by the *roll angle* ϕ, obtaining frame $\mathcal{F}_b$ by making axis y_b point towards the right of the airframe , Figure 2(c).

The rotation of the reference frame $\mathcal{F}_v$ to obtain reference frame $\mathcal{F}_b$ can also be characterized by an unit vector $\vec{\omega} = \omega_{x_v}\vec{u}_{x_v} + \omega_{y_v}\vec{u}_{y_v} + \omega_{z_v}\vec{u}_{z_v}$, identifying the axis of rotation, and an angle of rotation χ, Figure 2(d) [11]. The *quaternion* $Q_b^v = [q_0\ q_1\ q_2\ q_3]^T$ is defined in such a way that, $q_0 = \cos\frac{\chi}{2}$, $q_1 = \omega_{x1}\sin\frac{\chi}{2}$, $q_2 = \omega_{y1}\sin\frac{\chi}{2}$, and $q_3 = \omega_{z1}\sin\frac{\chi}{2}$. Quaternions are used to represent rotations of coordinate systems because they lead to simpler, polynomial, and singularity-free computations [11].

The quaternion Q_b^v can be obtained from the Euler angles as follows [11]:

$$Q_b^v \quad = \quad \begin{bmatrix} \cos\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} \\ \sin\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} - \cos\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} \\ \cos\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2} \\ \cos\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2} - \sin\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2} \end{bmatrix} \tag{1}$$

On the other hand,the Euler angles are obtained from the quaternion Q_b^v as:

$$\begin{array}{rcl} \phi & = & \arctan\left(2\frac{q_2q_3+q_0q_1}{1-2(q_1^2+q_2^2)}\right) \\ \theta & = & \arcsin\left(-2(q_1q_3-q_0q_2)\right) \\ \phi & = & \arctan\left(2\frac{q_1q_2+q_0q_3}{1-2(q_2^2+q_3^2)}\right) \end{array} \tag{2}$$

Let $\vec{v}_v$ be a vector in frame $\mathcal{F}_v$ and $\vec{v}_b$ be the same vector in frame $\mathcal{F}_b$. The two representations are related by $\vec{v}_b = R_b^v \vec{v}_v$, where R_b^v is the rotation matrix from frame $\mathcal{F}_v$ to frame $\mathcal{F}_b$. The inverse rotation matrix is the transpose $R_b^v = \left(R_v^b\right)^T$.

In terms of Euler angles, the rotation matrix has the form:

$$R_v^b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

and, in terms of the quaternion Q_b^v the rotation matrix is:

$$R_b^v = 2\begin{bmatrix} q_0^2+q_1^2-\frac{1}{2} & q_1q_2-q_0q_3 & q_1q_3+q_0q_2 \\ q_1q_2+q_0q_3 & q_0^2+q_2^2-\frac{1}{2} & q_2q_3-q_0q_1 \\ q_1q_3-q_0q_2 & q_2q_3+q_0q_1 & q_0^2+q_3^2-\frac{1}{2} \end{bmatrix} \tag{4}$$

Quaternions are used instead of Euler angles because they lead to simpler, polynomial, and singularity-free dynamical equations for the quadrotor [11]. Moreover, these polynomial dynamic equations are exploited in this paper to obtain alternative linearized systems for reachability analysis.

The nonlinear model for a quadrotor is defined by the following set of equations:

$$\begin{bmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{h} \end{bmatrix} = R_b^v \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv-qw \\ pw-ru \\ qu-pv \end{bmatrix} + R_v^b \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \frac{1}{m}\begin{bmatrix} 0 \\ 0 \\ -F \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{7}$$

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} qr(J_y-J_z)/J_x \\ pr(J_z-J_x)/J_y \\ pq(J_x-J_y)/J_z \end{bmatrix} + \begin{bmatrix} \tau_\phi/J_x \\ \tau_\theta/J_y \\ \tau_\psi/J_z \end{bmatrix} \tag{9}$$

Equation (5) is the *Translation Kinematics*, equation (6) is the *Translation Dynamics*, equations (7) and (8) represent the *Rotation Kinematics* in terms of Euler angles and quaternions, respectively, and equation (9) is the *Rotation Dynamics*. The above equations can be obtained by considering the physical principles of a quadrotor frame and then applying the Newton-Euler

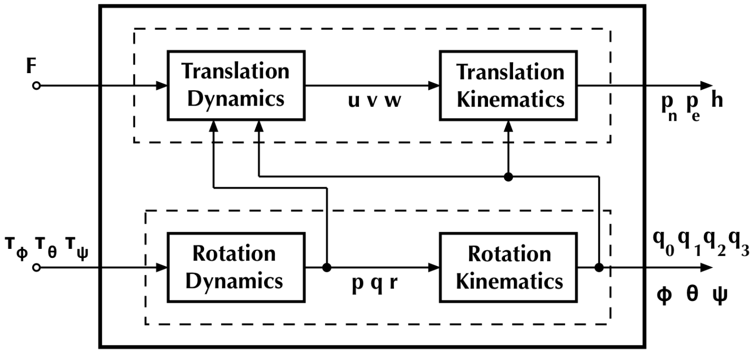


Figure 3: Model Block Diagram.

or the Lagrangian methods [1]. In Equations (5) to (9), m is the mass of the quadrotor, g is the acceleration of the gravity, and J_x, J_y, and J_z are the x_b, y_b, and z_b axis moments of inertia of the airframe, respectively. Finally, R_b^v is the rotation matrix from frame $\mathcal{F}_v$ to frame $\mathcal{F}_b$ and $R_b^v = \left(R_v^b\right)^T$ is the inverse rotation matrix. In the expanded version of the paper, we present the rotation matrices in terms of quaternions and Euler angles and also the unraveled dynamic equations.

Figure 3 shows a block diagram for the nonlinear model of a quadrotor, indicating the data flow of the variables along the blocks.

The nonlinear model expressed in terms of Euler angles is:

$$\begin{array}{rcl}
\dot{p}_n &=& w(\sin\phi\sin\psi + \cos\phi\cos\psi\sin\theta) + \\
&& -v(\cos\phi\sin\psi - \cos\psi\sin\phi\sin\theta) + u\cos\psi\cos\theta \\
\dot{p}_e &=& v(\cos\phi\cos\psi + \sin\phi\sin\psi\sin\theta) + \\
&& -w(\cos\psi\sin\phi - \cos\phi\sin\psi\sin\theta) + u\cos\theta\sin\psi \\
\dot{h} &=& w\cos\phi\cos\theta - u\sin\theta + v\cos\theta\sin\phi \\
\dot{u} &=& rv - qw - g\sin\theta \\
\dot{v} &=& pw - ru + g\cos\theta\sin\phi \\
\dot{w} &=& qu - pv - F/m + g\cos\phi\cos\theta \\
\dot{\phi} &=& p + r\cos\phi\tan\theta + q\sin\phi\tan\theta \\
\dot{\theta} &=& q\cos\phi - r\sin\phi \\
\dot{\psi} &=& r\cos\phi/\cos\theta + q\sin\phi/\cos\theta \\
\dot{p} &=& \tau_\phi/J_x + qr(J_y - J_z)/J_x \\
\dot{q} &=& \tau_\theta/J_y + pr(J_x - J_z)/J_y \\
\dot{r} &=& \tau_\psi/J_z + pq(J_x - J_y)/J_z
\end{array} \tag{10}$$

And the model expressed in terms of quaternions is:

$$\begin{aligned}
\dot{p}_n &= u(2q_0^2+2q_1^2-1)-v(2q_0q_3-2q_1q_2)+w(2q_0q_2+2q_1q_3)\\
\dot{p}_e &= v(2q_0^2+2q_2^2-1)+u(2q_0q_3+2q_1q_2)-w(2q_0q_1-2q_2q_3)\\
\dot{h} &= w(2q_0^2+2q_3^2-1)-u(2q_0q_2-2q_1q_3)+v(2q_0q_1+2q_2q_3)\\
\dot{u} &= rv-qw-g(2q_0q_2-2q_1q_3)\\
\dot{v} &= pw-ru+g(2q_0q_1+2q_2q_3)\\
\dot{w} &= qu-pv-F/m+g(2q_0^2+2q_3^2-1)\\
\dot{q}_0 &= -q_1p/2-q_2q/2-q_3r/2\\
\dot{q}_1 &= q_0p/2-q_3q/2+q_2r/2\\
\dot{q}_2 &= q_3p/2+q_0q/2-q_1r/2\\
\dot{q}_3 &= q_1q/2-q_2p/2+q_0r/2\\
\dot{p} &= \tau_\phi/J_x+qr(J_y-J_z)/J_x\\
\dot{q} &= \tau_\theta/J_y+pr(J_x-J_z)/J_y\\
\dot{r} &= \tau_\psi/J_z+pq(J_x-J_y)/J_z
\end{aligned} \tag{11}$$

Notice that while the Euler angles model is characterized trigonometric functions and singularities, the quaternion model is polynomial and singularity free.

2.2 Linearization

In order to perform control design and reachability analysis we use two approaches for the linearization of the quadrotor nonlinear equations.

First, we present the linearization around an equilibrium point, corresponding to the quadrotor being at a static hovering position in 3D space. The basic hypothesis are that $u_0 = v_0 = w_0 = 0\ m/s$ and $p_0 = q_0 = r_0 = 0\ rad/s$, and the following relations come from the analysis of the nonlinear equations: $\phi_0 = \theta_0 = \psi_0 = 0\ rad$, $F_0 = mg\ N$, $\tau_\phi = \tau_\theta = \tau_\psi = 0\ N.m$, and $(p_{n0}, p_{e0}, h_0) \in \mathbb{R}^3\ [m]$. The linearized model can be written as the following simple set of linear equations[3]:

$$\begin{array}{llllllllllll}
\dot{p}_n &=& u & \dot{u} &=& -g\theta & \dot{\phi} &=& p & \dot{p} &=& \tau_\phi/J_x\\
\dot{p}_e &=& v & \dot{v} &=& g\phi & \dot{\theta} &=& q & \dot{q} &=& \tau_\theta/J_y\\
\dot{h} &=& w & \dot{w} &=& -F/m & \dot{\psi} &=& r & \dot{r} &=& \tau_\psi/J_z
\end{array} \tag{12}$$

The corresponding model block diagram is depicted in Figure 4.

The previous linearized model is useful for control design, because of its simplicity. The decoupling of the variables is exploited in the controller design, in order to make nested control architecture. Notice that the rotational part of the nonlinear model presented in [9] is equivalent to the rotational part of the linear model in equation (12). The relevant difference in the translational parts of both models is that the linear velocities in [9] are expressed in terms of inertial frame coordinates.

When applying the reachability analysis to investigate the stability and the performance of the closed loop system, we need a richer model that could better represent the actual system behavior. One option could be to use the nonlinear model in Section 2.1. Notice that this nonlinear model has also uncertainties and unmodelled dynamics, like lift and drag forces [12], and would contribute with some of the possible traces of the state trajectory. Therefore, we propose a piecewise affine model for the quadrotor to be applied in the reachability analysis [4].

[3] We work the Euler angles model because it can be shown that the linearized quaternion model comes up with uncontrollable modes.

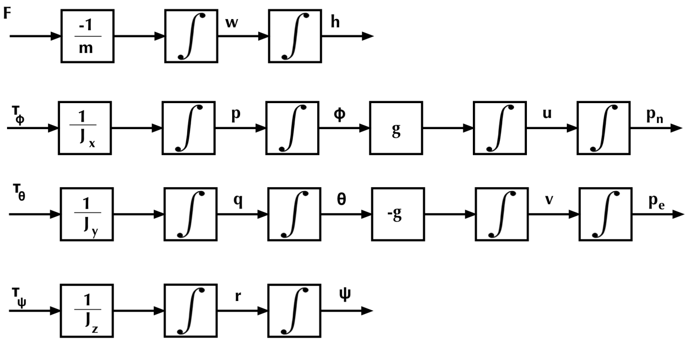


Figure 4: Linear Model Block Diagram (Euler angles).

Our model exploits the polynomial equations of the quaternions nonlinear model. Notice that the nonlinearities of the quaternion model are essentially quadratic and cubic monomials on the state variables. We use McCormick relaxations [7] to overapproximate these polynomial terms.

Given two variables defined within intervals with lower and upper bounds $x_1 \in [x_1^L, x_1^U]$ and $x_2 \in [x_2^L, x_2^U]$, the McCormick relaxation defines a set of planes for lower and upper bounds for the quadratic monomial $w_{12} = x_1 x_2$ as [7]:

$$\begin{array}{rcl} w_{12} & \leq & x_1^L x_2 + x_2^U x_1 - x_1^L x_2^U \\ w_{12} & \leq & x_1^U x_2 + x_2^L x_1 - x_1^U x_2^L \\ w_{12} & \geq & x_1^L x_2 + x_2^L x_1 - x_1^L x_2^L \\ w_{12} & \geq & x_1^U x_2 + x_2^U x_1 - x_1^U x_2^U \end{array} \tag{13}$$

The above inequalities define a convex polyhedron in the space defined by x_1, x_2 and w_{12}, that overapproximates the values for w_{12}.

Figure 5 illustrates the convex polyhedron that overapproximates the quadratic monomial by the planes defined by the McCormick relaxation.

As an illustration, we use the McCormick relaxation planes to create a polyhedron to overapproximate the quadratic monomials that appear in the rotation dynamics part of the quadrotor nonlinear model, equation (9). Specifically we approximate the products pr, qr and pq in those equations.

We define the following set of equations as a linear approximation for the rotation dynamics:

$$\begin{array}{rcl} \dot{p} & = & \tau_\phi / J_x + B_1 (J_y - J_z)/J_x \\ \dot{q} & = & \tau_\theta / J_y + B_2 (J_x - J_z)/J_y \\ \dot{r} & = & \tau_\psi / J_z + B_3 (J_x - J_y)/J_z \end{array} \tag{14}$$

where variables B_1, B_2 and B_3 are overappoximations for the quadratic monomials qr, pr and pq in (9) obtained using McCormick relaxations [7].

We've provided bounds to p, q and r by observing the result of various simulation scenarios and considering the dimensions of the *CrazyFlie*. From this, we've arbitrated that $p^L = q^L = r^L = -1.0\ rad/s$ and $p^U = q^U = r^U = 1.0\ rad/s$. We are working on SpaceEx components

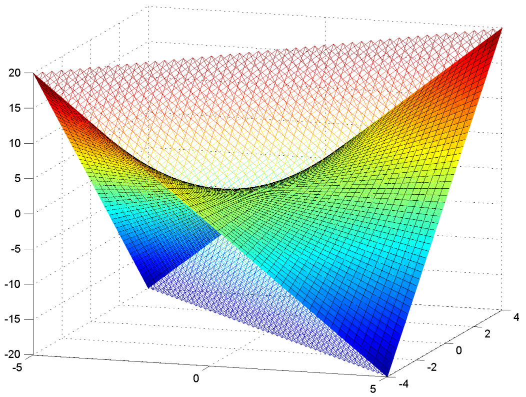


Figure 5: Function $w_{12} = x_1 x_2$ and McCormick relaxation convex polyhedron for $x_1^L = -5.0$, $x_2^L = -4.0$, $x_1^U = 5.0$ and $x_2^U = 4.0$.

based on McCormick relaxations to overapproximate other polynomial terms that appear in the quaternions nonlinear model.

2.3 Controller Design

The strapdown sensors available in the *CrazyFlie* are a rate gyro, an accelerometer, a magnetometer and a barometer. Using sensor fusion we can obtain with certain precision the angular rates, the quaternion, the Euler angles, the linear speeds, and the height [1, 12]. We disregard the horizontal position information for the moment.

The *CrazyFlie* has an open-source firmware, where the control and sensor fusion routines can be programmed [13]. Communications with the *CrazyFlie* are performed through a computer running the *CrazyFlie PC Client* and with the *CrazyRadio dongle* installed. Pilot commands can be sent to the *CrazyFlie* through a game controller connected to the computer. A recent version of the *CrazyFlie* also has IOS / Android clients for remote piloting.

We've chosen as reference inputs for the attitude controller the height and the orientation of the quadrotor. These references can can be generated by the game controller [13].

The architecture of the attitude controller is shown in Figure 6. We've proposed a nested control architecture. The inner loops are the *Angular Rates* and the *Vertical Speed* controllers that act directly on the thrust F and the torques τ_ϕ, τ_θ and τ_ψ. In an outer loop, the *Vertical Position* and the *Angular Orientation* controllers receive the references to the height and orientation, and transform them into references for the inner loop controllers.

The proposed nested control architecture permits the development of each control loop in a relatively independent way, being each controller responsible for a different aspect of the overall attitude control. It is inspired on both the decoupling of the linear model presented in Section 2.2 and the common practice of aircraft and missile control design [2].

Finally, the *CrazyFlie* firmware permits the processing of the control/sensor fusion routines and the update of the outputs at an rate up to $500Hz$. Being conservative, we've chosen the update rate for the attitude controller at $100Hz$. With this, we've arbitrated the eigenvalues

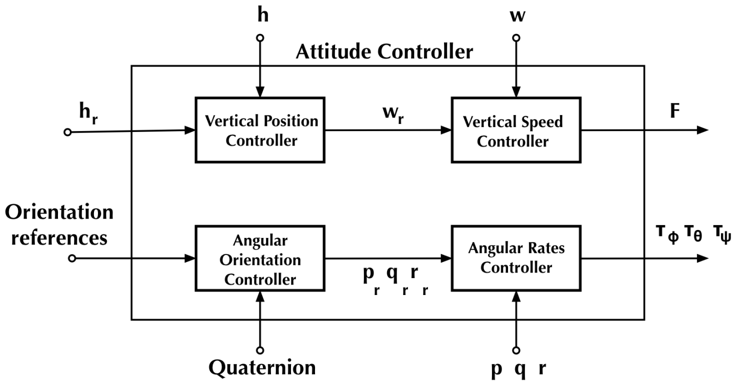
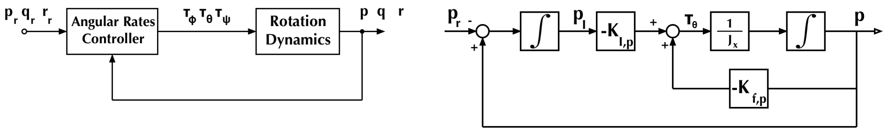


Figure 6: Attitude Controller Architecture.



(a) Design perspective.

(b) Robust tracking.

Figure 7: Angular Rates Controller.

of the inner loop controllers to be at 1/5 to 1/8 of the controller update frequency, and the eigenvalues for the outer loop controllers at 1/5 to 1/8 of the inner loop controllers.

We illustrate the steps to design the *Angular Rates Controller*, depicted in Figure 7(a). By the decoupling of the nonlinear model, Section 2.1, the *Angular Rates Controller* acts directly on the rotation dynamics. Also, by considering the decoupling of the variables in the linearized model, Section 2.2, we decompose the *Angular Rates Controller* into three loops, one for the roll rate p, one for the pitch rate q, and one for the yaw rate r. Figure 7(b) shows the general scheme for the roll rate loop with robust tracking and integral control [14]. We've selected the poles for the closed loop system to be at $(-\sqrt{2}/2 \pm i\sqrt{2}/2) \times 20$ and we've computed the controller gains $K_{I,p}$ and $K_{f,p}$ by pole placement.

Table 1 shows the obtained values for the controller gains. The other controllers, for the pitch and yaw rates, were synthesized by an equivalent procedure. Therefore, the angular rates controller has the corresponding control law

$$\begin{array}{rcl} \tau_\phi & = & -K_{I,p}p_I - K_{f,p}p \\ \tau_\theta & = & -K_{I,q}q_I - K_{f,q}q \\ \tau_\psi & = & -K_{I,r}r_I - K_{f,r}r \end{array} \tag{15}$$

and as state variables

$$\begin{array}{rcl} \dot{p}_I & = & p - p_r \\ \dot{q}_I & = & q - q_r \\ \dot{r}_I & = & r - r_r \end{array} \tag{16}$$

Table 1: Controller Gains.

Loop i	$K_{I,i}$	$K_{f,i}$
p	3.6144×10^{-3}	2.5557×10^{-4}
q	3.6507×10^{-3}	2.5814×10^{-4}
r	7.7472×10^{-3}	5.4781×10^{-4}

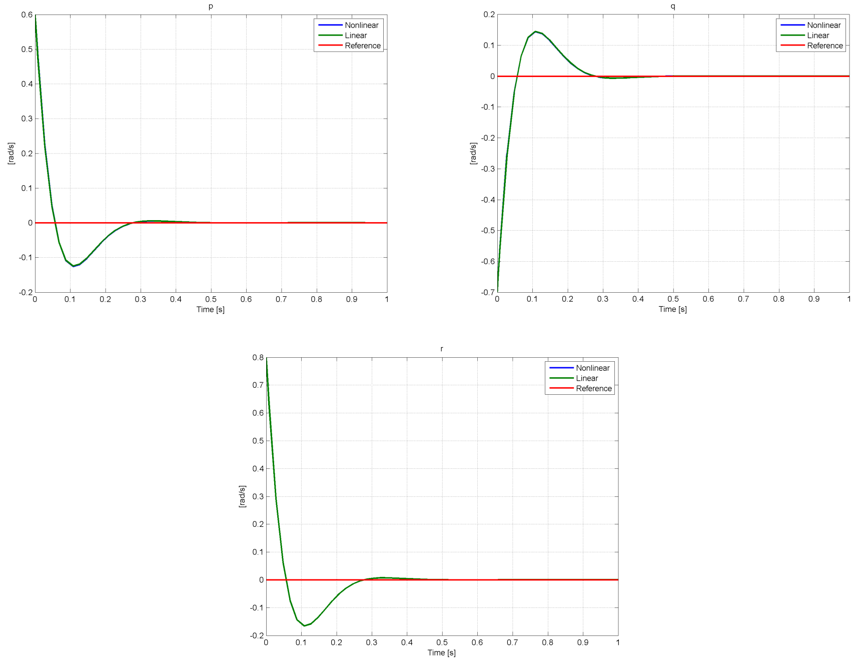


Figure 8: Simulation Results.

where p_r, q_r and r_r are the reference inputs and p_I, q_I and r_I are the integral states.

We illustrate the performance of the angular rates controller by means of simulations. The simulation has as initial values for $[p\ q\ r]^T$ as $[0.6\ \ -0.7\ 0.8]^T\ rad/s$ and as reference inputs $[0\ 0\ 0]^T rad/s$. The results are shown in Figure 8. The linear velocities reach the desired values with a settling time of less than $0.5s$. There is an overshoot of 20% and the linear and the nonlinear systems present very similar behaviors.

3 Key Observations

In this section we present the guidelines and some preliminary results for the application of reachability analysis in the design of the attitude controller for the quadrotor.

We intend to use reachability analysis techniques to investigate and improve the robustness of the controller. The analysis is performed using the tool SpaceEx [4][4], that is a scalable tool

[4] http://spaceex.imag.fr/

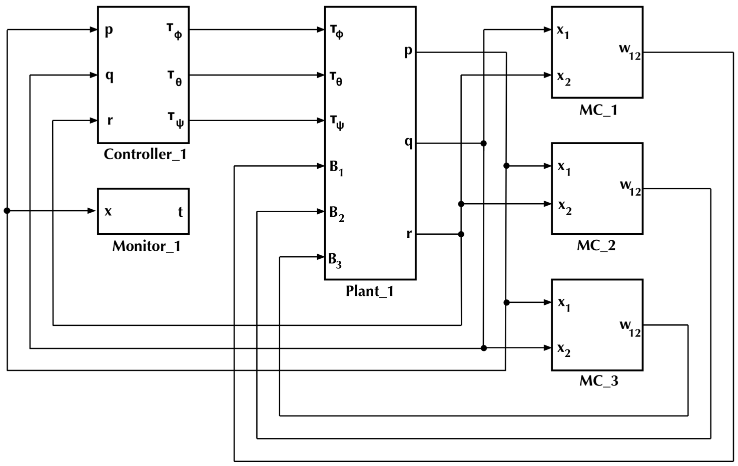


Figure 9: Reachability Analysis Block Diagram.

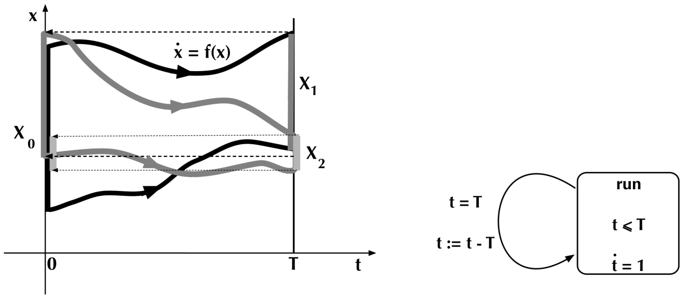
for the reachability analysis for piecewise affine systems.

Figure 9 shows the block diagram for the proposal of application of the reachability analysis to the angular rates controller. Components *Plant_1* and *MC_1* to *MC_3* implement the piecewise affine model detailed in Section 2.2, where *MC_i* implements the McCormick approximation for variable B_i in equation (14), for $i \in \{1, 2, 3\}$. Component *Controller_1* implements the angular rates controller presented in Section 2.3.

Component *Monitor_1* implements a monitor, a component used in reachability analysis to define a bounded time interval and a safety limits to the variables.

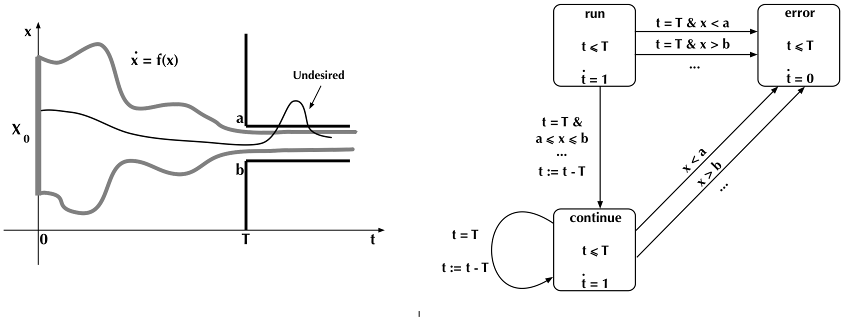
The main idea to address infinite time horizon properties in control system design by means of reachability analysis, where a finite time horizon T is to be provided by the user, is to use a monitor automaton with guard $t = T$ and reset $t := t - T$, as in Figure 10. At each such discrete transition, the reachability analysis algorithm tests if the resulting state space X_j is contained in any of the precedent reached state spaces X_k. If it is the case, the computation terminates. In this case, termination corresponds to a boundedness proof. Moreover, if the the initial set of states contains an equilibrium point, termination means that the system is stable.

We can improve the results of the reachability analysis by assigning bad states to the monitor, as in Figure 11. The monitor component in Figure 11(b) takes state x as input and has t as a controlled variable. If state *error* of the monitor is not reachable, we can reason that all the system trajectories remain within a distance, parametrized by bounds a and b, from the *setpoint* in an infinite time horizon. State *run*, transitions from *run* to *error* and from *run* to *continue* are to check if the flowpipe attains the target given by $a \leq x \leq b$ in a finite time horizon T. State *continue* and its outgoing transitions to *error* are to check if no trajectory remains in the bound after reaching at the target in $t = T$, as in the pointed trajectory. We can reason on values for T and on the bounds a and b from practical observation or knowledge



(a) Forward reachability illustration. (b) Monitor.

Figure 10: Forward reachability and simple monitor.



(a) Improving forward reachability (b) Improved monitor.

Figure 11: Forward reachability with guards and monitor.

of the system. Particularly, *Monitor_1* in Figure 9 takes care of bounds for variable p.

As a first illustration, the reachability analysis of the angular rates controller is performed with the following set of initial conditions: $-1 \leq p \leq 1$, $-1 \leq q \leq 1$ and $-1 \leq r \leq 1$. Termination of the reachability analysis has occurred, therefore the controller shows to be stable and robust to the variations in the plant. A graphical illustration of the result of the reachability analysis is shown in Figure 12.

In Figure 13, the flow pipes for the state trajectories of p, q and r are shown, now with initial conditions $0.5 \leq p \leq 0.7$, $-0.8 \leq q \leq -0.6$, and $0.7 \leq r \leq 0.9$.

The polynomial quaternions nonlinear model can also be exploited by other reachability analysis techniques, like the ones based on Bernstein polynomials [5]. Flight envelopes as in [9] can also be applied for the safety controller design using the quaternions model. Moreover, in [6] the problem of the region of attraction of polynomial dynamic systems is addressed.

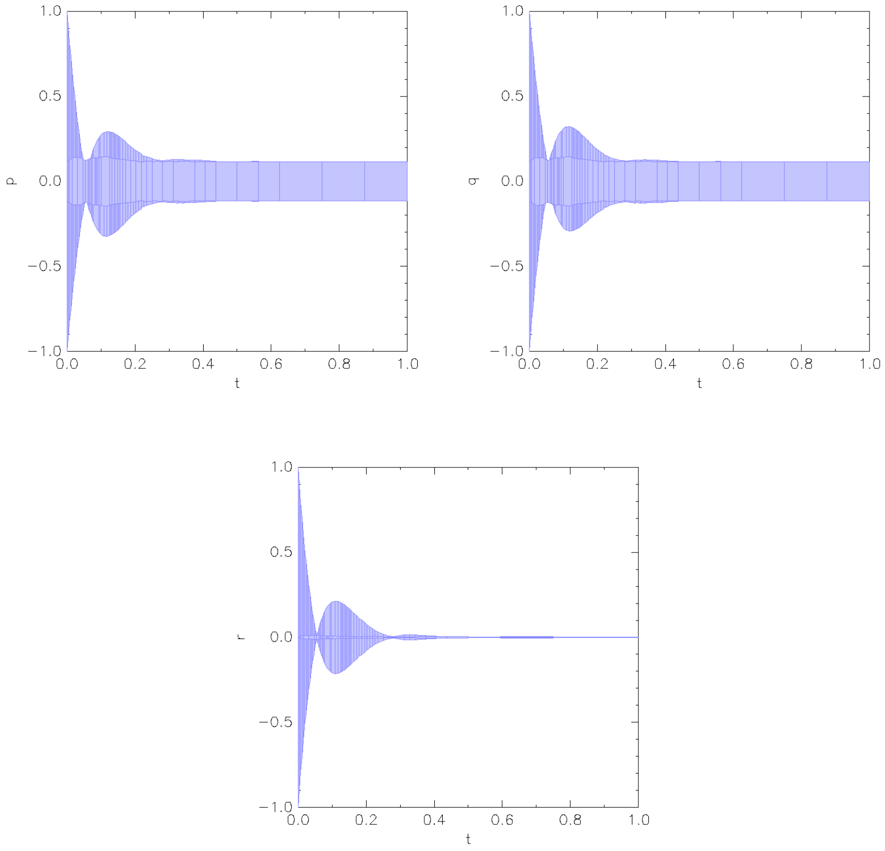


Figure 12: Reachability Analysis Results.

4 Outlook

In the following steps we will complete the attitude controller design, using the reachability analysis as a control design support tool. Then we will implement the controller in the *CrazyFlie* and perform measurements of the closed loop system. In a further step, we plan to perform measurements and enhance the proposed piecewise affine model using system identification techniques [8]. In a possible expansion, a reported dependence of the actuators performance on the battery voltage could be exploited in a varying parameter approach [15].

Acknowledgments

The author gratefully thanks the valuable comments and suggestions gently given by Goran Frehse and Oded Maler, from VERIMAG, while preparing this paper.

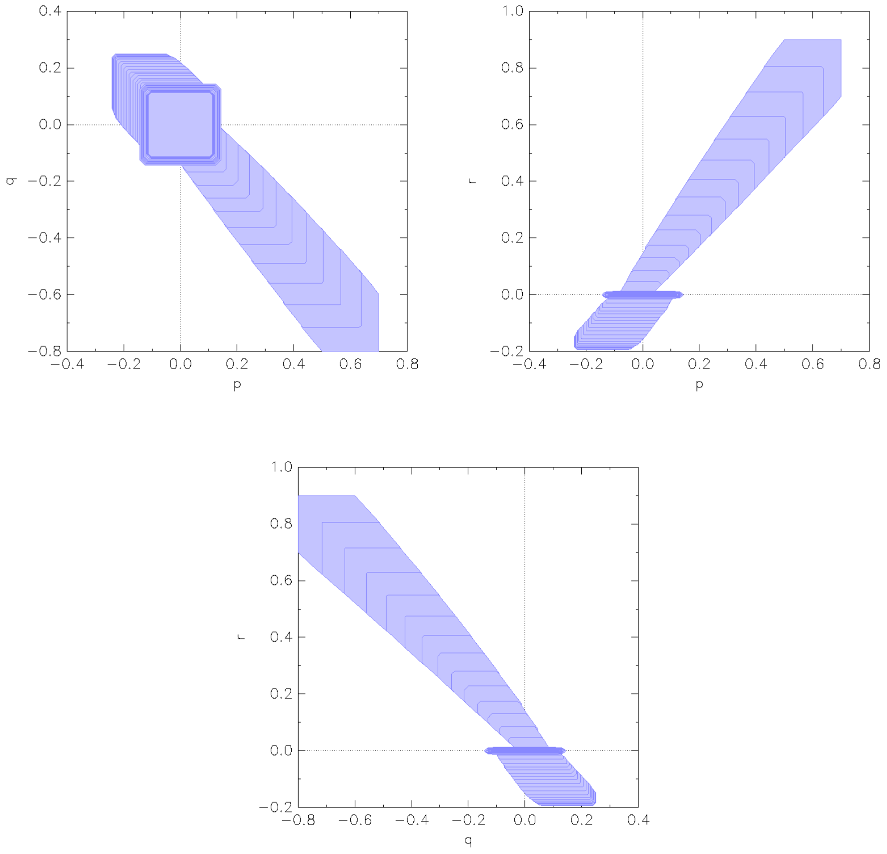


Figure 13: Flow pipes.

References

[1] R. W. Beard, “Quadrotor dynamics and control,” Brigham Young University, Tech. Rep., 2008.

[2] J. H. Blackelock, *Automatic Control of Aircraft and Missiles*, 2nd ed. John Wiley & Sons, 1991.

[3] O. Maler, “Algorithmic verification of continuous and hybrid systems,” in *INFINITY*, ser. EPTCS, L. Holík and L. Clemente, Eds., vol. 140, 2014, pp. 48–69.

[4] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler, “Spaceex: Scalable verification of hybrid systems,” in *Proc. 23rd International Conference on Computer Aided Verification (CAV)*, ser. LNCS, S. Q. Ganesh Gopalakrishnan, Ed. Springer, 2011.

[5] T. Dang and R. Testylier, “Reachability analysis for polynomial dynamical systems using the bernstein expansion,” *Reliable Computing Journal, Special Issue: Bernstein Polynomials*, 2012.

[6] D. Henrion and M. Korda, “Convex computation of the region of attraction of polynomial control systems.” *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 297–312, 2014.

[7] F. A. Al-Khayyal and J. E. Falk, “Jointly constrained biconvex programming,” *Mathematics of Operations Research*, vol. 8, no. 2, pp. 273–286, May 1983.

[8] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal, “Identification of hybrid systems: A tutorial,” *European Journal of Control*, vol. 13, pp. 242–260, 2007.

[9] S. Kaynama and C. J. Tomlin, “Benchmark: Flight envelope protection in autonomous quadrotors,” in *Benchmarks of the Applied Verification for Continuous and Hybrid Systems (ARCH)*, 2014, available at http://cps-vo.org/node/12114.

[10] S. Bouabdallah, “Design and control of quadrotors with application to autonomous flying,” Thése Docteur ès Sciences, Faculté des Sciences et Techniques de l’Ingénieur, École Polytechnique Fédérale de Lausanne (EPFL), Februrary 2007.

[11] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*, 2nd ed., ser. IEE Radar, Sonar, Navigation and Avionics. The Institution of Electrical Engineers, 2004, vol. 17.

[12] R. Leishman, J. Macdonald, R. Beard, and T. McLain, “Quadrotors and accelerometers: State estimation with an improved dynamic model,” *Control Systems, IEEE*, vol. 34, no. 1, pp. 28–41, Feb 2014.

[13] Bitcraze, “Bitcraze wiki,” Available *on line* at https://wiki.bitcraze.io/, last checked at August 28, 2015, 2015.

[14] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 6th ed. Pearson Education Inc. and Prentice Hall, 2010, iSBN 9780136019695.

[15] W. Hanna, “Modelling and control of an unmanned aerial vehicle,” Monography for Bachelor of Engineering degree, School of Engineering & Information Technology, Faculty of Engineering, Health, Science and Environment, Charles Darwin University, 2014.