Quadcopters Full Quaternions BackStepping Control

1st Alberto Petrucci
L'Aquila, Italy
petrucci.alberto@gmail.com

2nd Costanzo Manes
L'Aquila, Italy
costanzo.manes@univaq.it

3rd Alessandro D'Innocenzo
L'Aquila, Italy
alessandro.dinnocenzo@univaq.it

***Abstract*—Quadcopters are a great way to practice control and observers theory. In this paper the nonlinear dynamic model of the quadcopter is formulated using the Newton-Euler method and, in particular, the attitude is represented by a unit-quaternion. All the quadcopter's model can be divided in two dynamics: the rotational dynamic (attitude motion) and the translational dynamic (x, y and z motion). A nonlinear position control technique based on the well known $BeckStepping$-control is presented and verified by Lyapunov methods. All the simulations were done using MATLAB and the results have been compared with a minimal control energy technique: the LQR. In addition everything has been verified in reality by implementing control on the CrazyFlie 2.0 quadcopter.**

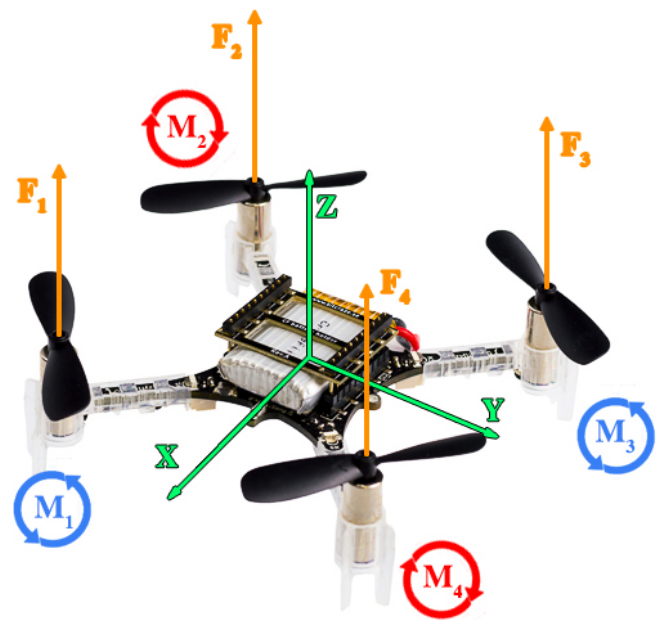
***Keywords*—Quadcopter, UAV, Nonlinear Control, Backstepping, LQR, Quaternions**

I. Introduction

In the last year UAVs have seen a growing research interest due to the large kind of applications that is possible to realize with them. In this paper we present a robust nonlinear control with disturbance estimation and compensation for quadcopters. This kind of control is able to make the flight stable and secure (all the tests were done indoor). We describe the translational and rotational dynamics combining Newton-Euler formalism with quaternions. It's important to underline that quaternions reduce the computational cost and mathematical errors given by the approximation introduced by the *Look-Up-Table* needed to implement the main trigonometric functions. The main contribution of our work lies in developing an architecture that is able to increase the robustness of the controller thanks to the estimation of the disturbances given by two main factors: model uncertainties and real external disturbance.

II. Related Work

There are a different work centered on the full control of UAVs; however, different approaches have been proposed during the year. In [11] has been proposed a control technique based on the Integral backstepping approach and a mathematical model based on Euler-Angles. In [5] and [6] a command filtered backstepping approach has been described. The main advantages of command filtered backstepping is that you can simplify the controller obviating the need for analytical computation of signal derivatives. Other researchers use a optimal nonlinear control approach that minimize the system energy and so increase the drone's flight time. This can be seen in [8] and [9]. Finally the last control approach is that related to the geometric tracking control like SO(3) that is presented in [12]. In contrast with this paper we increase the robustness of the controller thanks to a disturbance observer.



III. Quadcopter Dynamic Model

The quadcopter model will be developed based on a Newton-Euler approach with the following assumptions (according [1]):

- The quadcopter structure is rigid and symmetrical
- The center of gravity of the quadcopter coincides with the vehicle frame origin
- The propellers are rigid
- Thrust force and drag torque are proportional to the square of propeller's speed.

First of all its necessary to define two reference systems: the world and the inertial ones. The following figure represent the inertial frame and we can note that the quadcopter is in formation "X". At the beginning of a generic test the only difference between the two systems is a simple translation and so all the x, y and z axes are parallel. So, to get a generic rotation from the two system (with quaternions) is really useful report the following matrix:

$$\mathbf{R}(\mathbf{q}) = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_0q_3+q_1q_2) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix}$$

where $\mathbf{q}$ is a generic unit-quaternion defined like:

$$\mathbf{q} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T \tag{1}$$

and q_0 is the quaternion angle; instead $\mathbf{q_{1:3}}$ is the quaternion axis. So, the first part of the quadcopter dynamic can be fully characterized by:

- the quaternions dynamics
- the rotational speed dynamics

The first is given by the equations:

$$\dot{q} = \frac{1}{2}\begin{bmatrix} -q_1 & -q_3 & -q_4 \\ q_0 & -q_4 & +q_3 \\ q_3 & +q_1 & -q_2 \\ -q_2 & +q_2 & +q_1 \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{2}$$

instead the rotational speed dynamics is directly connected to the inertia matrix and to the moments with the equations:

$$\dot{\omega} = \begin{bmatrix} -\omega_y\omega_z \\ \omega_x\omega_z \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{1}{md^2} & 0 & 0 \\ 0 & \frac{1}{md^2} & 0 \\ 0 & 0 & \frac{1}{2md^2} \end{bmatrix}\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} \tag{3}$$

where M_x, M_y and M_z are the moments applied to the x, y and z axis; m is the mass of the quadcopter and d is the length of the quadcopter arms multiplied by the $sin(45deg)$ because the quadcopter is in "X" formation.
We now analyze the transnational dynamic. We have considered a model with as input the total thrust force along the z axis so it can be described by the equations:

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \\ \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ 0 \\ 0 \\ -g \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{2q_1q_3+2q_0q_2}{m} \\ \frac{2q_2q_3-2q_0q_1}{m} \\ \frac{q_0^2-q_1^2-q_2^2+q_3^2}{m} \end{bmatrix} F_t \tag{4}$$

where $g = 9.81$ is the gravity value and the second vector simply breaks down the total thrust $F_t = F_1 + F_2 + F_3 + F_4$ in its components along the x, y and z axes.
To complete the model we have to introduce the link between the input vector and the motor's forces. This is given by the matrix:

$$u = \begin{bmatrix} F_t \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} F_1 + F_2 + F_3 + F_4 \\ K_d(-F_1 - F_2 + F_3 + F_4) \\ K_d(-F_1 + F_2 + F_3 - F_4) \\ K_c(-F_1 + F_2 - F_3 + F_4) \end{bmatrix} \tag{5}$$

A. The full quaternion model

From a control point of view it is really useful to have a complete model characterized by a unique state space. In this case the state vector is defined in this way:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ \omega_x \\ \omega_y \\ \omega_z \\ p_x \\ p_y \\ p_z \\ v_x \\ v_y \\ v_z \end{bmatrix} \tag{6}$$

so we have that the complete quadcopter model is described by the equation:

$$\dot{X} = F(X) + G(X)u \tag{7}$$

where

$$F = \begin{bmatrix} -\frac{x_2x_5+x_3x_6+x_4x_7}{2} \\ \frac{x_1x_5-x_4x_6+x_3x_7}{2} \\ \frac{x_4x_5+x_1x_6-x_2x_7}{2} \\ \frac{-x_3x_5+x_2x_6+x_1x_7}{2} \\ -x_6x_7 \\ x_5x_7 \\ 0 \\ x_{11} \\ x_{12} \\ x_{13} \\ 0 \\ 0 \\ -g \end{bmatrix} \tag{8}$$

and

$$G = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{md^2} & 0 & 0 \\ 0 & 0 & \frac{1}{md^2} & 0 \\ 0 & 0 & 0 & \frac{1}{2md^2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{2x_2x_4+2x_1x_3}{m} & 0 & 0 & 0 \\ \frac{2x_3x_4-2x_1x_2}{m} & 0 & 0 & 0 \\ \frac{x_1^2-x_2^2-x_3^2+x_4^2}{m} & 0 & 0 & 0 \end{bmatrix} \tag{9}$$

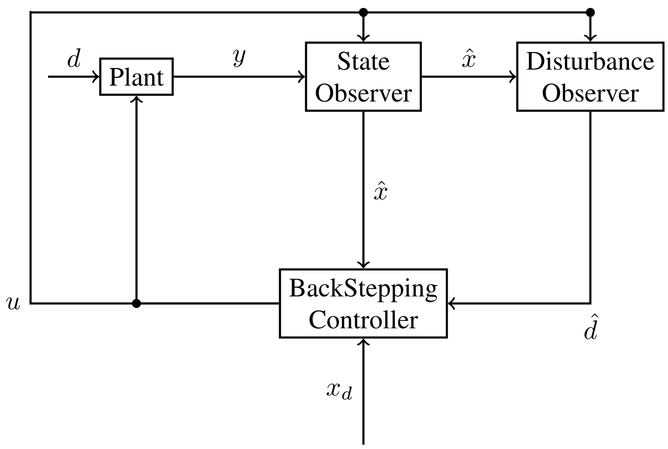
so we have to note that the model is nonlinear and affine.

IV. Full Quaternion BackStepping Controller

In this section we will show how to apply BackStepping methods to find a nonlinear controller for the quadcopter. It is important to underline that in general with a controller we are able to directly control a number of state variables equal to the number of inputs. In this case we are particularly fortunate because all the inputs are decoupled and so we can compute the control-inputs one by one without the use of a matrix inversion. We decided to directly control the following state variables:

$$\begin{bmatrix} x_2 & x_3 & x_4 & x_{10} \end{bmatrix} \quad (10)$$

With this choice we can indirectly control the angle of the unit-quaternion. Furthermore we can indirectly control the components of the thrust along x and y axis simply changing the quaternion axis. So, in general, the main control loop can be rapresented by the following diagram:



The state observer is an EKF presented in [2]. The disturbance observer has been computed using a Levant differentiator. As we can see the controller receives as inputs:

- the optimal state estimation $\hat{x}$
- the estimation, $\hat{d}$, of the disturbance d
- the desired state x_d

A. The BackStepping controller

We can start to show the Lyapunov stability of the controller related to x_{10}.Considering a Lyapunov function like:

$$V_{10} = \frac{1}{2} e_{10}^2 \quad (11)$$

where $e_{10} = x_{d10} - x_{10}$ and x_{d10} is the desired altitude. We can compute its derivative:

$$\dot{V}_{10} = e_{10}\dot{e}_{10} = e_{10}(\dot{x}_{d10} - \dot{x}_{10}) = e_{10}(\dot{x}_{d10} - x_{13}) \quad (12)$$

now we have to show that $\dot{V}_{10} \le 0$ so, choosing a definite positive function:

$$P_{10}(e) = c_{10}e_{10}^2, \quad c_{10} > 0 \quad (13)$$

It must be verified that:

$$\dot{V}_{10} < -P_{10}(e) \quad (14)$$

Set

$$x_{d13} = \dot{x}_{d10} + c_{10}e_{10} \quad (15)$$

and

$$e_{13} = x_{13} - \dot{x}_{d10} - c_{10}e_{10} \quad (16)$$

Adding and subtracting $e_{10}c_{10}e_{10}$ in the equation 12 we obtain that:

$$\begin{aligned} \dot{V}_{10} &= e_{10}(\dot{x}_{d10} - x_{13} + c_{10}e_{10} - c_{10}e_{10}) \\ &= e_{10}(-e_{13} - c_{10}e_{10}) \\ &= -e_{10}e_{13} - c_{10}e_{10}^2 \end{aligned}$$

So, considering the Lyapunov function

$$V_{13} = V_{10} + \frac{1}{2}e_{13}^2 \quad (17)$$

we have

$$\begin{aligned} \dot{V}_{13} &= \dot{V}_{10} + e_{13}\dot{e}_{13} \\ &= -e_{10}e_{13} - c_{10}e_{10}^2 + e_{13}(\dot{x}_{13} - \ddot{x}_{d10} - c_{10}\dot{e}_{10}) \end{aligned}$$

and so replacing the terms related to $\dot{x}_{13}$ and $\dot{e}_{10}$ we obtain:

$$\begin{aligned} \dot{V}_{13} = -e_{10}e_{13} - c_{10}e_{10}^2 + e_{13}(-g + \frac{x_1^2 - x_2^2 - x_3^2 + x_4^2}{m}F_t \\ - \ddot{x}_{d10} - c_{10}(\dot{x}_{d10} - x_{13})) \end{aligned}$$

setting now

$$\begin{aligned} F_t = \frac{m}{x_1^2 - x_2^2 - x_3^2 + x_4^2}(g + e_{10} + \ddot{x}_{d10} - c_{13}e_{13} \\ + c_{10}(\dot{x}_{d10} - x_{13})) \end{aligned}$$

we obtain

$$\dot{V}_{13} = -c_{10}e_{10}^2 - c_{13}e_{13}^2 \le 0 \quad (18)$$

We can show the same property for the virtual input $U_x = 2x_2x_4 + 2x_1x_3$ (for l'$U_y = 2x_3x_4 - 2x_1x_2$ all the steps will be really similar):

$$\begin{aligned} V_8 &= \tfrac{1}{2}e_8^2 \\ \dot{V}_8 &= e_8\dot{e}_8 = e_8(\dot{x}_{d8} - x_{11}) \end{aligned} \quad (19)$$

set

$$\begin{aligned} x_{d11} &= \dot{x}_{d8} + c_8e_8 \\ e_{11} &= x_{11} - \dot{x}_{d8} - c_8e_8 \end{aligned} \quad (20)$$

adding and subtracting $e_8c_8e_8$ into the $\dot{V}_8$ we obtain:

$$\begin{aligned} \dot{V}_8 &= e_8(\dot{x}_8 - x_{11} + c_8e_8 - c_8e_8) \\ &= e_8(-e_{11} - c_8e_8) \\ &= -e_8e_{11} - c_8e_8^2 \end{aligned}$$

Choosing the following Lyapunov function

$$V_{11} = V_8 + \frac{1}{2}e_{11}^2 \quad (21)$$

we got that its derivative will be

$$\dot{V}_{11} = -e_8e_{11} - c_8e_8^2 + e_{11}(\dot{x}_{11} - \ddot{x}_{d8} - c_8\dot{e}_8) \quad (22)$$

and so

$$\dot{V}_{11} = -e_8e_{11} - c_8e_8^2 + e_{11}(-\ddot{x}_{d8} - c_8(\dot{x}_{d8} - x_{11}) + \frac{2}{m}U_xF_t) \quad (23)$$

Choosing

$$U_x = \frac{m}{2F_t}(e_8 + \ddot{x}_{d8} - c_{11}e_{11} + c_8(\dot{x}_{d8} - x_{11})) \quad (24)$$

we obtain that $\dot{V}_{11} \leq 0$ and so that the related controlled dynamic will be asymptotically stable.
Now we'll present the most relevant part of the controller: the quaternion's axis controller.
Recalling that our attitude is rapresented by the quaternion

$$q = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T \tag{25}$$

using the Lyapunov theorem
Setting

$$\begin{aligned} e_2 &= x_{d2} - x_2 \\ e_3 &= x_{d3} - x_3 \\ e_4 &= x_{d4} - x_4 \end{aligned} \tag{26}$$

ad choosing the following functions

$$\begin{aligned} V_2 &= \tfrac{1}{2}e_2^2 \\ V_3 &= \tfrac{1}{2}e_3^2 \\ V_4 &= \tfrac{1}{2}e_4^2 \end{aligned} \tag{27}$$

we can compute their temporal derivative

$$\begin{aligned} \dot{V}_2 &= e_2\dot{e}_2 \\ &= e_2(\dot{x}_{d2} - \dot{x}_2) \\ &= e_2[\dot{x}_{d2} - \frac{1}{2}(x_1x_5 - x_4x_6 + x_3x_7)] \end{aligned} \tag{28}$$

$$\begin{aligned} \dot{V}_3 &= e_3\dot{e}_3 \\ &= e_3(\dot{x}_{d3} - \dot{x}_3) \\ &= e_3[\dot{x}_{d3} - \frac{1}{2}(x_4x_5 + x_1x_6 - x_2x_7)] \end{aligned} \tag{29}$$

$$\begin{aligned} \dot{V}_4 &= e_4\dot{e}_4 \\ &= e_4(\dot{x}_{d4} - \dot{x}_4) \\ &= e_4[\dot{x}_{d4} - \frac{1}{2}(-x_3x_5 + x_2x_6 + x_1x_7)] \end{aligned} \tag{30}$$

From here, repeating the previous reasoning, we can define the following quantities:

$$\begin{aligned} \bar{x}_{d2} &= \dot{x}_{d2} + c_2e_2 \\ \bar{x}_{d3} &= \dot{x}_{d3} + c_3e_3 \\ \bar{x}_{d4} &= \dot{x}_{d4} + c_4e_4 \\ \bar{e}_2 &= \tfrac{1}{2}(x_1x_5 - x_4x_6 + x_3x_7) - \dot{x}_{d2} - c_2e_2 \\ \bar{e}_3 &= \tfrac{1}{2}(x_4x_5 + x_1x_6 - x_2x_7) - \dot{x}_{d3} - c_3e_3 \\ \bar{e}_4 &= \tfrac{1}{2}(-x_3x_5 + x_2x_6 + x_1x_7) - \dot{x}_{d4} - c_4e_4 \end{aligned}$$

adding and subtracting into (28), (29) and (30) respectly

$$\begin{aligned} e_2c_2e_2 \\ e_3c_3e_3 \\ e_4c_4e_4 \end{aligned} \tag{31}$$

we got:

$$\begin{aligned} \dot{V}_2 &= e_2[\dot{x}_{d2} - \frac{1}{2}(x_1x_5 - x_4x_6 + x_3x_7) + c_2e_2 - c_2e_2] \\ &= -e_2\bar{e}_2 - c_2e_2^2 \end{aligned}$$

$$\begin{aligned} \dot{V}_3 &= e_3[\dot{x}_{d3} - \frac{1}{2}(x_4x_5 + x_1x_6 - x_2x_7) + c_3e_3 - c_3e_3] \\ &= -e_3\bar{e}_3 - c_3e_3^2 \end{aligned}$$

$$\begin{aligned} \dot{V}_4 &= e_4[\dot{x}_{d4} - \frac{1}{2}(-x_3x_5 + x_2x_6 + x_1x_7) + c_4e_4 - c_4e_4] \\ &= -e_4\bar{e}_4 - c_4e_4^2 \end{aligned}$$

So, choosing the Lyapunov functions:

$$\begin{aligned} \bar{V}_2 &= V_2 + \tfrac{1}{2}\bar{e}_2^2 \\ \bar{V}_3 &= V_3 + \tfrac{1}{2}\bar{e}_3^2 \\ \bar{V}_4 &= V_4 + \tfrac{1}{2}\bar{e}_4^2 \end{aligned} \tag{32}$$

whose time derivatives are respectively:

$$\begin{aligned} \dot{\bar{V}}_2 &= \dot{V}_2 + \bar{e}_2\dot{\bar{e}}_2 \\ \dot{\bar{V}}_3 &= \dot{V}_3 + \bar{e}_3\dot{\bar{e}}_3 \\ \dot{\bar{V}}_4 &= \dot{V}_4 + \bar{e}_4\dot{\bar{e}}_4 \end{aligned} \tag{33}$$

that is

$$\begin{aligned} \dot{\bar{V}}_2 = {} & -e_2\bar{e}_2 - c_2e_2^2 + \bar{e}_2[-\ddot{x}_{d2} - c_2\dot{e}_2 \\ & + \frac{1}{2}(\dot{x}_1x_5 + x_1\dot{x}_5 - \dot{x}_4x_6 - x_4\dot{x}_6 + \dot{x}_3x_7 + x_3\dot{x}_7)] \end{aligned}$$

$$\begin{aligned} \dot{\bar{V}}_3 = {} & -e_3\bar{e}_3 - c_3e_3^2 + \bar{e}_3[-\ddot{x}_{d3} - c_3\dot{e}_3 \\ & + \frac{1}{2}(\dot{x}_4x_5 + x_4\dot{x}_5 + \dot{x}_1x_6 + x_1\dot{x}_6 - \dot{x}_2x_7 + x_2\dot{x}_7)] \end{aligned}$$

$$\begin{aligned} \dot{\bar{V}}_4 = {} & -e_4\bar{e}_4 - c_4e_4^2 + \bar{e}_4[-\ddot{x}_{d4} - c_4\dot{e}_4 \\ & + \frac{1}{2}(-\dot{x}_3x_5 - x_3\dot{x}_5 + \dot{x}_2x_6 + x_2\dot{x}_6 + \dot{x}_1x_7 + x_1\dot{x}_7)] \end{aligned}$$

so, replacing the time derivative of x_i with their respective equations defined in 8, 9 and 5, we obtain three different equations in the unknown M_x, M_y and M_z. Solving the system so that:

$$\begin{aligned} \dot{\bar{V}}_2 &= -c_2e_2^2 - \bar{c}_2\bar{e}_2^2 \\ \dot{\bar{V}}_3 &= -c_3e_3^2 - \bar{c}_3\bar{e}_3^2 \\ \dot{\bar{V}}_4 &= -c_4e_4^2 - \bar{c}_4\bar{e}_4^2 \end{aligned} \tag{34}$$

we obtain that $\dot{\bar{V}}_i \leq 0$, $i = 2, 3, 4$ and so that the related dynamics are asymptotically stable. To compute the control inputs can be really useful the use of the symbolic calculation. From here to show the global asymptotic stability we can consider the following Lyapunov function

$$V_t = \tfrac{1}{2}e^Te, \quad e = \begin{bmatrix} e_2 \\ e_3 \\ e_4 \\ e_8 \\ e_9 \\ e_{10} \end{bmatrix} \tag{35}$$

its derivative is:

$$\dot{V}_t = e^T \begin{bmatrix} -e_{11} - c_8e_8 \\ -e_{12} - c_9e_9 \\ -e_{13} - c_{10}e_{10} \\ -\bar{e}_2 - c_2e_2 \\ -\bar{e}_3 - c_3e_3 \\ -\bar{e}_4 - c_4e_4 \end{bmatrix} \tag{36}$$

choosing now

$$\bar{V}_t = V_t + \frac{1}{2}\bar{e}^T\bar{e}, \quad \bar{e} = \begin{bmatrix} \bar{e}_2 \\ \bar{e}_3 \\ \bar{e}_4 \\ e_{11} \\ e_{12} \\ e_{13} \end{bmatrix} \tag{37}$$

whose derivative is:

$$\dot{\bar{V}}_t = \dot{V}_t + \bar{e}^T\dot{\bar{e}} = \dot{V}_t + \bar{e}^T \begin{bmatrix} \dot{e}_{11} \\ \dot{e}_{12} \\ \dot{e}_{13} \\ \dot{\bar{e}}_2 \\ \dot{\bar{e}}_3 \\ \dot{\bar{e}}_4 \end{bmatrix} \tag{38}$$

Using the precedent equations for the inputs (F_t, M_x, M_y and M_z) we get that $\dot{\bar{V}}_t \leq -c_t e^T e \leq 0$ with $c_t > 0$.

B. Disturbance Observer

We have supposed that the drone is affected by an unknown disturbances that can directly affect the linear and angular acceleration dynamics. So, they are linear forces or moments along x, y and z axes. To do this was necessary to generalize the model introducing the following matrix:

$$D_F = \frac{1}{m}\begin{bmatrix} 0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0\\1&0&0\\0&1&0\\0&0&1 \end{bmatrix} \quad D_M = \begin{bmatrix} 0&0&0\\0&0&0\\0&0&0\\0&0&0\\ \frac{1}{md^2}&0&0\\0&\frac{1}{md^2}&0\\0&0&\frac{1}{2md^2}\\0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0 \end{bmatrix} \tag{39}$$

So, the full model is always affine but with the following form:

$$\dot{X} = F(X) + G(X)u + D_F d_f + D_M d_m \tag{40}$$

$$d_f = \begin{bmatrix} F_{dx} \\ F_{dy} \\ F_{dz} \end{bmatrix} \quad d_m = \begin{bmatrix} M_{dx} \\ M_{dy} \\ M_{dz} \end{bmatrix} \tag{41}$$

where d_f and d_m are the unknown disturbances. We have developed a disturbance observer, based on Levant differentiator, that is able to give an estimation of d_f and d_m. It's important to underline that the controller is able to compensate all the estimated disturbances and that all the demonstrations made before are still valid because the only difference is that the inputs are affected by unknowns disturbances terms.

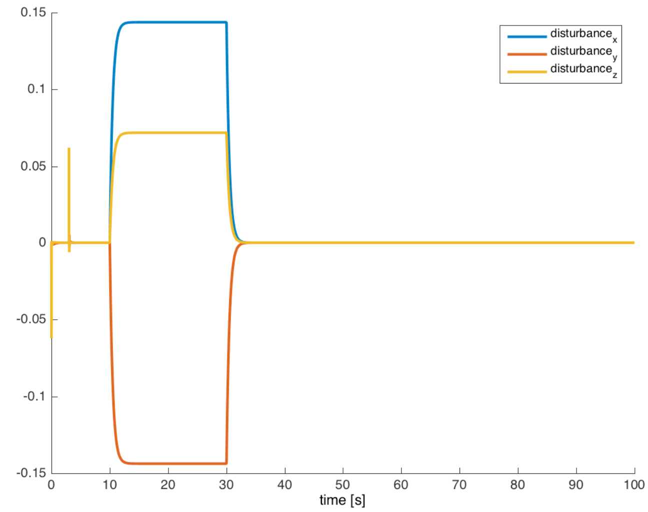


Fig. 1. Force disturbance estimation along axes

C. Trajectory Tracking

With the previous control law we are able to follow any 3D path using these strategies:

- we can control the z position simply choosing an altitude reference x_{10}
- we can control the x and y coordinates indirectly changing the quaternion axis reference

We define the desired attitude quaternion, q_d, from the previous virtual inputs U_x and U_y in this way:

$$q_d = \frac{1}{||\tilde{q}_d||}\tilde{q}_d \quad \text{and} \quad \tilde{q}_d = \begin{bmatrix} \alpha & \tilde{q}_{d1} & \tilde{q}_{d2} & \tilde{q}_{d3} \end{bmatrix}^T \tag{42}$$

where

$$\begin{bmatrix} \alpha \\ \tilde{q}_{d1} \\ \tilde{q}_{d2} \\ \tilde{q}_{d3} \end{bmatrix} = \begin{bmatrix} \geq 1 \\ -\frac{U_y - x_3 * x_4}{x_1} \\ \frac{U_x - x_2 * x_4}{x_1} \\ x_{d4} \end{bmatrix} \tag{43}$$

It's important to note that if $\alpha > 1$ then the convergence will be slower and that if x_{d4} is constant then any trajectory will be followed with a fixed heading. So computing the quaternion error

$$q_e = q^{-1} \otimes q_d \tag{44}$$

we are able to find our attitude reference.

V. Simulations

In the following section we'll present a simulation in which we compared our controller with an LQR controller. The simulation consists in tracking a circular path along the xy

plane (with a radius of $r = 1m$ and a pulse $\omega = 1$). We supposed that the quadcopter initial state is

$$X = \begin{bmatrix}1\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix} = \begin{bmatrix}q_0\\q_1\\q_2\\q_3\\\omega_x\\\omega_y\\\omega_z\\p_x\\p_y\\p_z\\v_x\\v_y\\v_z\end{bmatrix} \tag{45}$$

The following figures represent the evolution of the state trajectory.

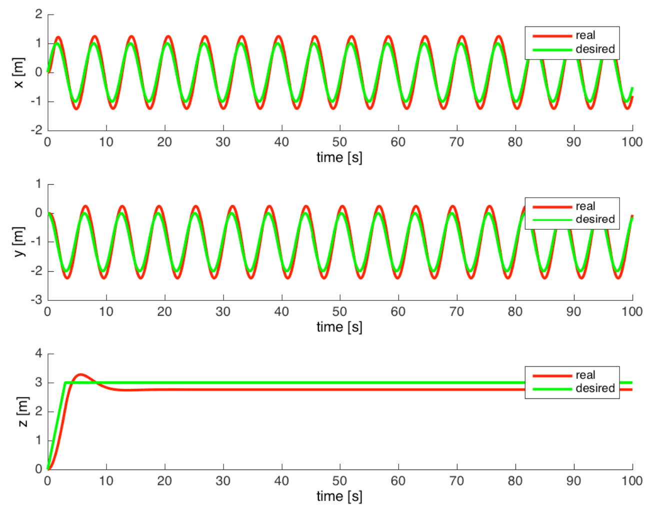


Fig. 2. LQR simulation

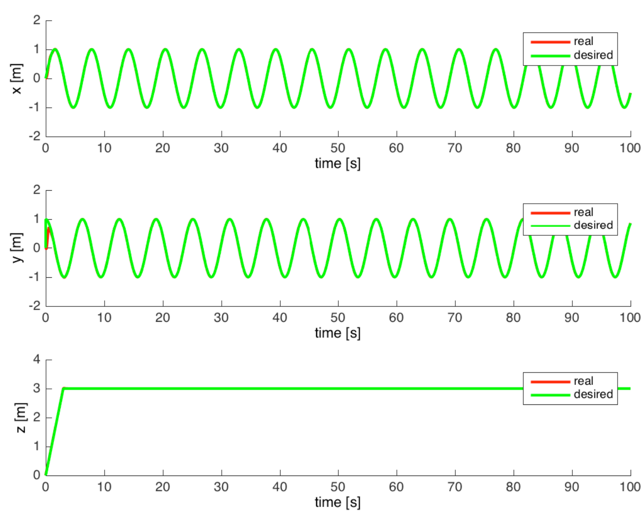


Fig. 3. BackStepping simulation

In the first one we can see that there is a constant error during the evolution. This is mainly caused by two different factors:

- the rapid dynamic of the path
- the model linearization needed by the LQR controller

In the second figure instead the convergence is really fast and the quadcopter is able to follow the path with really a small error.

VI. Conclusions

In this paper we presented a mathematical model of a quadcopter fully based on quaternions. Than we have developed a position control algorithm based on the backstepping. We showed that the inputs found with this method make the system globally asymptotically stable. After that we have simulated the quadcopter model on MATLAB and we have compared our control algorithm with the LQR. Finally we implemented the controller on a real quadcopter (the CrazyFlie 2.0).

References

[1] R. W. Beard, “Quadrotor dynamics and control,” 2008.

[2] Mueller, Mark W and Hamer, Michael and D’Andrea Raffaello, Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadrocopter state estimation, 2015 IEEE International Conference on Robotics and Automation (ICRA),2015, May, pp.1730-1736.

[3] Arie Levant, *Robust exact differentiation*, 2004.

[4] Miki Livne, Arie Levant, *Proper discretization of homogeneous differentiators*, 2014.

[5] Stefano Di Lucia, Gian Diego Tipaldi, Wolfram Burgard “Attitude Stabilization Control of an Aerial Manipulator using a Quaternion-Based Backstepping Approach,” 2015 IEEE

[6] Sheng Zhao, Wenjie Dong, Jay A. Farrell, “Quaternion-based Trajectory Tracking Control of VTOL-UAVs using Command Filtered Backstepping,” 2013 AACC

[7] Emil Fresk, George Nikolakopoulos, “Full Quaternion Based Attitude Control for a Quadrotor,” 2013 EUCA

[8] Kaddouri Djamel, Mokhtari Abdellah, Abdelaziz Benallegue, “Attitude Optimal Backstepping Controller Based Quaternion for a UAV,” Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2016, Article ID 8573235

[9] Wesam Jasim, Dongbing Gu, “H_∞ Path Tracking Control for Quadrotors Based on Quaternion Representation,” Springer International Publishing Switzerland 2014

[10] Xing Huo, Mingyi Huo, Hamid Reza Karimi, “Attitude Stabilization Control of a Quadrotor UAV by Using Backstepping Approach,” Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2014, Article ID 749803

[11] Samir Bouabdallah, Roland Siegwart, “Full Control of a Quadrotor,” 2007 IEEE

[12] Taeyoung Lee, Melvin Leok, N. Harris McClamroch, “NONLINEAR ROBUST TRACKING CONTROL OF A QUADROTOR UAV ON SE(3),” 2012 John Wiley and Sons Asia Pte Ltd and Chinese Automatic Control Society