ABSTRACT
A Fractional Order Sliding Mode Control (FOSMC) is proposed for improving a quadrotor’s flight: take-off, hover, navigation and landing. The control scheme is based on previous works on FOSMC for quadrotors, where a fractional derivative is used to improve the quadrotor’s behaviour in presence of uncertainties and disturbances. A comparison between an Integer Order Sliding Mode Control (IOSMC) scheme and the Fractional Order Sliding Mode Control (FOSMC) scheme is made.

KEY WORDS
Fractional Control, Fractional Calculus, Fractional Derivative, Fractional Order Sliding Mode Control

1 Introduction
Nowadays quadrotors are used in a variety of applications such as recreational purposes to surveillance, inspection, recognition and rescue. The dynamic model of a quadrotor consist of twelve first order non-linear differential equations with six degrees of freedom and only four control inputs, which makes the control strategy for such a system a challenge [13]. The quadrotor vertical take off and landing ability has the advantage of no need of landing track [2]. The quadrotor is a complete mechatronic system, that involves mechanics, electronics, aerodynamics and computing issues [12].

On the other hand “The Sliding Mode Control strategy has been successfully applied in a wide variety of design problems, from the movement control systems and chemical processes to the control of chaotic systems. A wide range of applications in the domain of Sliding Mode Control with robust properties, motivate the design of low level laws for quadrotors”[1]. Moreover Fractional Control and Modelling is an old new science, with more than 300 years old, but recently it has focus attention in many engineering and science applications. “Fractional calculus (which includes fractional integration and fractional differentiation) is as old as its familiar counterpart, classical calculus (or integer order calculus). For quite a long time it developed slowly. However, in the past few decades, fractional calculus has attracted increasing interest due to its applications in science and engineering”[5].

The abstract definition of fractional derivatives is a beautiful mathematical entity which represent the theory background for this work, but we can not directly use these definitions in simulation, neither in real experimental results, thus numeric methods approaches exist to implement a fractional derivative. In this work the so called Grünwald-Letnikov numeric method for fractional derivatives is used.

In this paper a FOSMC is proposed for improving a Quadrotor’s flight, it is claimed that a FOSMC has some robust properties as in [1]. To show this issue disturbances are introduced to the model of the quadrotor, mainly parametric disturbances. A FOSMC strategy is designed for the desired roll and pitch angles, as well as for the main thrust control.

The article is organized as follows. In section 2, the model of the quadrotor is described. In section 3 a brief introduction to fractional calculus is presented, as well as the FOSMC strategy. In section 4 some simulation results are presented. Finally in section 5 some conclusions are given.

2 Quadrotor’s Dynamic Model
Let us consider an inertial reference frame and a body fixed reference frame to specify the position, velocity and acceleration of the quadrotor. The world frame W, is defined by axes X, Y and Z, with Z pointing upward. The body frame B, xB, yB, zB, is attached to the center of mass of the quadrotor. A picture of a quadrotor with the corresponding reference frames is shown in Figure 1.

The Newton-Euler approach was used to obtain the dynamical behaviour of the quadrotor [3], which considers a quadrotor as a rigid body. Let us consider the generalized coordinates of the quadrotor denoted as \( q = (\xi, \eta) \), where \( \xi = (x, y, z) \) is the position vector, which goes from the center of mass of the quadrotor relative to the fixed reference frame \((X, Y, Z)\), and \( \eta = (\phi, \theta, \psi) \) are the Euler angles of the quadrotor. These angles have the following names and they represent a rotation around their corre-
Figure 1. Coordinate Systems and moment/forces acting on the quadrotor.

The corresponding axis: $\phi$ is the pitch angle around xB axis, $\theta$ is the roll angle around yB axis, and $\psi$ is the yaw angle around zB axis. The angular velocity vector $\Omega = (p, q, r)$ is related to $\dot{\eta}$ by the equivalent form $\Omega = W_\eta \dot{\eta}$ where

$$
W_\eta = \begin{bmatrix}
1 & 0 & -\sin(\theta) \\
0 & \cos(\phi) & \cos(\theta)\sin(\phi) \\
0 & -\sin(\phi) & \cos(\theta)\cos(\phi)
\end{bmatrix}
$$

is a standard kinematic matrix. Also $R(\phi, \theta, \psi)cSO(3)$, is the so called rotation matrix that represents the quadrotor’s orientation that is relative to the fixed reference frame.

Using the compact notation $c(\alpha) = \cos(\alpha)$ and $s(\alpha) = \sin(\alpha)$. This matrix is given by

$$
R = \begin{bmatrix}
c(\theta)c(\psi) & c(\psi)s(\phi) - c(\phi)s(\psi) & s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta) \\
c(\phi)c(\psi) + s(\phi)s(\theta)s(\psi) & c(\theta)c(\psi) & c(\phi)s(\theta)s(\psi) - c(\phi)c(\psi)s(\theta) \\
-s(\phi)c(\psi) & c(\phi)c(\psi)s(\theta) - c(\psi)s(\theta) & c(\phi)s(\psi)
\end{bmatrix}
$$

Additionally the inertia matrix is given by

$$
I = \begin{bmatrix}
I_{xx} & 0 & 0 \\
0 & I_{yy} & 0 \\
0 & 0 & I_{zz}
\end{bmatrix}
$$

where $I_{xx}$, $I_{yy}$ and $I_{zz}$ are the moments of inertia of the quadrotor about its corresponding axis, $m$ is the mass of the quadrotor and $g = 9.81[\text{m/s}^2]$ is the gravitational constant.

Finally, the dynamics of a rigid body under external forces take the form:

$$
\dot{\zeta} = \begin{bmatrix}
v_x \\ v_y \\ v_z
\end{bmatrix}, 
$$

$$
m\ddot{\zeta} = R \begin{bmatrix}
0 \\ 0 \\ T_f
\end{bmatrix} - mg, 
$$

$$
\ddot{R} = R\dot{\Omega}, 
$$

$$
I\ddot{\Omega} = -\Omega \times I\Omega + \tau, 
$$

where $\dot{\Omega}$ is the anti-symmetric matrix of $\Omega$, also called the wedge operator defined as

$$
\dot{\Omega} = \begin{bmatrix}
0 & -r & q \\
r & 0 & -p \\
-q & p & 0
\end{bmatrix}
$$

and $\times$ denotes the cross product. $T_f$ is the total thrust applied to the quadrotor, this is $T_f = \sum_{i=1}^{4} f_i = k \sum_{i=1}^{4} \omega_i^2$ and represents the first control signal, furthermore $\tau = (\tau_\phi, \tau_\theta, \tau_\psi)$ is the vector of control signals for roll, pitch and yaw angles respectively.

From equations (1)-(4) one can obtain the dynamics of the quadrotor that describe its motion. These equations are the following ones [3],[10]:

$$
\dot{x} = \frac{\cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi)}{m} u_1, 
$$

$$
\dot{y} = \frac{\cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi)}{m} u_1, 
$$

$$
\dot{z} = -g + \frac{\cos(\phi)\cos(\theta)}{m} u_1, 
$$

$$
\dot{\phi} = \dot{\psi} \left(\frac{I_{xx} + I_{yy} - I_{zz}}{I_{xx}}\right) + \frac{u_2}{I_{xx}}, 
$$

$$
\dot{\theta} = \dot{\psi} \left(\frac{I_{zz} - I_{xx} - I_{yy}}{I_{yy}}\right) + \frac{u_3}{I_{yy}}, 
$$

$$
\dot{\psi} = \dot{\phi} \left(\frac{I_{zz} - I_{xx} - I_{yy}}{I_{zz}}\right) + \frac{u_4}{I_{zz}}, 
$$

where $u_1 = T_f$, $u_2 = \tau_\phi$, $u_3 = \tau_\theta$ and $u_4 = \tau_\psi$ are considered as control signals.

It is possible to stablish a mapping between the control signals $u_1$, $u_2$, $u_3$ and $u_4$ and the angular velocity of each quadrotor’s electric motor $\omega_1$, $\omega_2$, $\omega_3$ and $\omega_4$. In fact, each motor produces a vertical force $F_i$ and a moment $M_i$ that is related to the motor angular velocity $\omega_i$ in the form [8]:

$$
F_i = k_F\omega_i^2, \quad M_i = k_M\omega_i^2, 
$$

where $k_F$ and $k_M$ are positive constants and can be determined experimentally. The relation between the motor angular velocities and the control signals is then given by [8]:

$$
\begin{bmatrix}
\omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4
\end{bmatrix} = K_{MF} \begin{bmatrix}
\epsilon_1^2 \\ \epsilon_2^2 \\ \epsilon_3^2 \\ \epsilon_4^2
\end{bmatrix}, 
$$

with $K_{MF}$ being an invertible matrix of the form

$$
K_{MF} = \begin{bmatrix}
k_F & k_F & k_F & k_F \\ 0 & k_F L & 0 & -k_F L \\ -k_F L & 0 & k_F L & 0 \\ k_M & -k_M & k_M & -k_M
\end{bmatrix}, 
$$

where $L$ is the distance from the rotation axis of the motors to the center of the quadrotor.
3 Fractional Order Sliding Mode Control

Fractional calculus is mainly the calculus of derivatives and integrals of fractional order that can be real or complex[4] and it has many applications in science and engineering. In this article, a Sliding Mode Control of Fractional Order is proposed for the trajectory tracking of a quadrotor. Among the definitions of fractional derivative the most cited are the Riemann-Liouville and Caputo definitions. Here, the Caputo definition is used which is expressed as:

$$D^\alpha \phi = \frac{1}{\Gamma(n-\alpha)} \int_0^t \frac{\phi^{(n)}(\xi)}{(t-\xi)^{\alpha+n-1}} d\xi,$$  
(14)

where $\Gamma$ is the so-called gamma function defined as $\Gamma(n) = \int_0^\infty t^{n-1}e^{-t} dt$ and $n-1 \leq \alpha \leq n$ being an integer. From this definition, one can notice that when $\alpha < 0$ one has integration and when $\alpha > 0$ one has differentiation; when $\alpha = 1.0$ one has the first order derivative of a function. The gamma function generalizes the notion of a number raised to an arbitrary real number (or complex number). Thus the gamma function somehow involves the computation of a fractional derivative. The definition of a fractional derivative, as stated above, can not be used in practice, thus numeric methods such as the one based on the Grünwald-Letnikov approach is commonly used.

In order to obtain the expression for the control signal $u_1$ the tracking error in $z$ is defined as $e_z = z - z_d$, where $z_d$ is a reference signal for $z$. Then a so-called switching function $s_z$ is chosen as:

$$s_z = \dot{e}_z + \lambda_z e_z,$$  
(15)

where $\lambda_z$ is a real constant parameter. The switching function $s_z$ defines the sliding surface

$$s_z = 0 = \dot{e}_z + \lambda_z e_z.$$  
(16)

Thus, $\lambda_z$ is selected in such a way that the first order linear differential equation (16) has a solution that exponentially converges to zero. Based on the strategy proposed in [1] to attract the dynamics of $z$ to the sliding surface (16), the $(1+\beta)$ fractional order derivative of $s_z$ is set to be

$$s_z^{1+\beta} = -\sigma_z sgn(s_z) - \mu_z s_z^{(\beta)},$$  
(17)

where $\sigma_z$ and $\mu_z$ are positive real constants. The derivative of (17) to the order $(-\beta)$ is now taken (notice that this is equivalent to integrate (17) to the order $\beta$), leading to

$$\dot{s}_z = -\sigma_z D^{-\beta} sgn(s_z) - \mu_z s_z.$$  
(18)

On the other hand, from (15) and (7) one has that

$$s_z = \ddot{z} - \dot{z}_d + \lambda_z e_z = -g + \cos(\phi)\cos(\theta)\frac{u_1}{m} - \dot{z}_d + \lambda_z e_z.$$  
(19)

Thus, the control signal $u_1$ takes the form

$$u_1 = \frac{m}{\cos(\phi)\cos(\theta)}(g + P_z),$$  
(20)

where

$$P_z = \ddot{z}_d - \lambda_z e_z - \sigma_z D^{-\beta} sgn(s_z) - \mu_z s_z.$$  
(21)

To compute the control signals $u_2$ and $u_3$, one first needs to define the desired reference signals for the angle $\phi$ and $\theta$, this is $\phi_d$ and $\theta_d$. For doing this, it is assumed that $\phi \approx 0$, $\theta \approx 0$ and $\psi \approx 0$, thus the dynamics (5), (6), (8) and (10) take the form

$$\dot{x} \approx \tan(\theta)(g + P_z),$$  
(22)

$$\dot{y} \approx -\tan(\phi)(g + P_z),$$  
(23)

$$\dot{\phi} = \frac{u_2}{I_{xx}}, \quad \dot{\theta} = \frac{u_3}{I_{yy}}, \quad \dot{\psi} = \frac{u_4}{I_{zz}}.$$  
(24)

One now considers that $\tan(\phi)$ and $\tan(\theta)$ are "virtual" input signals, this is $u_x = \tan(\theta)$ and $u_y = \tan(\phi)$. Similar to the previous reasoning for the computation of $u_1$, the tracking errors $e_x$ and $e_y$ are defined as $e_x = x - x_d$ and $e_y = y - y_d$, respectively, where $x_d$ and $y_d$ are reference signals for the signals $x$ and $y$. Then the following switching functions are chosen:

$$s_x = \dot{e}_x + \lambda_x e_x,$$  
(25)

$$s_y = \dot{e}_y + \lambda_y e_y,$$  
(26)

where $\lambda_x$ and $\lambda_y$ are real constant parameters selected in such a way that the first order linear differential equations defined by the sliding surfaces

$$s_x = 0 = \dot{e}_x + \lambda_x e_x,$$  
(27)

$$s_y = 0 = \dot{e}_y + \lambda_y e_y,$$  
(28)

have solutions that exponentially converge to zero. As before, and based on the strategy proposed in [1], the $(1+\beta)$ fractional order derivative of $s_x$ and $s_y$ are set as

$$s_x^{1+\beta} = -\sigma_x sgn(s_x) - \mu_x s_x^{(\beta)},$$  
(29)

$$s_y^{1+\beta} = -\sigma_y sgn(s_y) - \mu_y s_y^{(\beta)},$$  
(30)

where $\sigma_x$, $\sigma_y$, $\mu_x$, and $\mu_y$ are positive real constants. The derivative of (29) and (30) to the order $(-\beta)$ leads to the equations

$$\dot{s}_x = -\sigma_x D^{-\beta} sgn(s_x) - \mu_x s_x,$$  
(31)

$$\dot{s}_y = -\sigma_y D^{-\beta} sgn(s_y) - \mu_y s_y.$$  
(32)

From equations (22), (23), (27) and (28) one also has that

$$\dot{s}_x = \dot{e}_x + \lambda_x e_x = \tan(\theta)(g + P_z) - \dot{x}_d + \lambda_x e_x,$$  
(33)

$$\dot{s}_y = \dot{e}_y + \lambda_y e_y = \tan(\phi)(g + P_z) - \dot{y}_d + \lambda_y e_y.$$  
(34)

The desired signals $\theta_d$ and $\phi_d$ can then be computed from (33) and (34) when the signals $\tan(\theta)$ and $\tan(\phi)$ are considered to be virtual input signals, as mentioned before. This consideration allows to have the following expressions for $\theta_d$ and $\phi_d$
\[ \theta_d = \arctan \left( \frac{P_x}{P_x + g} \right), \]  
\[ \phi_d = -\arctan \left( \frac{P_y}{P_y + g} \right), \]  

where

\[ P_x = \ddot{x}_d - \lambda_x \dot{e}_x - \sigma_x D^{-\beta} \text{sgn}(s_x) - \mu_x s_x, \]  
\[ P_y = \ddot{y}_d - \lambda_y \dot{e}_y - \sigma_y D^{-\beta} \text{sgn}(s_y) - \mu_y s_y. \]

Notice now that, when \( \theta \) and \( \phi \) are managed to approach \( \theta_d \) and \( \phi_d \) as soon as possible, the FOSMC strategy discussed above allows to reach the sliding surface \( s_x = 0 \) and \( s_y = 0 \) in finite time where the tracking errors \( e_x \) and \( e_y \) tend to zero exponentially. In this paper, a simple proportional and derivation action is implemented to allow that \( \theta \rightarrow \theta_d \), \( \phi \rightarrow \phi_d \) and \( \psi \rightarrow \psi_d \); this is

\[ u_2 = I_{xx} \left[ \ddot{\theta}_d - K_d \dot{e}_\theta - K p_\theta e_\theta \right], \]  
\[ u_3 = I_{yy} \left[ \ddot{\phi}_d - K_d \dot{e}_\phi - K p_\phi e_\phi \right], \]  
\[ u_4 = I_{zz} \left[ -K d \dot{\psi} - K p_\psi \dot{e}_\psi \right], \]

where \( e_\phi = \phi - \phi_d, e_\theta = \theta - \theta_d \) and \( e_\psi = \psi - \psi_d \) are the tracking errors in the angles \( \phi, \theta \) and \( \psi \) respectively. Notice in particular, that \( \phi_d \) i set to 0 in (41).

4 Simulation Results

Some simulations were carried out in order to evaluate the performance of the control strategy proposed. The following simulations were made in MATLAB and using the quadrotor’s model described by equations (5)-(10). These dynamic equations are integrated within a script in MATLAB that implements a Runge-Kutta method of fourth order. On the other hand the Grünwald-Letnikov numeric method is used for the computation of the fractional control feedbacks (20), (39)-(41), a fractional order of \( \alpha = 0.5 \) is used, because it shows the best performance. The sign function is implemented using the expression:

\[ \text{sgn}(s_z) = \frac{s_z}{|s_z| + \epsilon} \]

where the small quantity \( \epsilon > 0 \) is a real positive constant.

A piecewise trajectory was used as a tracking reference in the space as follows: the first 5 seconds are used to the take-off of the quadrotor until it reachesHover at 1 meter, then a lemniscata 2 meters long is followed one time, with a period of 30 seconds per return, finally landing from 1 meter to the floor is performed within the last 5 seconds. The model parameters used in the simulation are shown in Table 1 while the control parameters are given in Table 2.

The trajectory of the quadrotor in the space is shown in Figure 2, where one can notice the effect of the disturbance.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass</td>
<td>1.4</td>
<td>kg</td>
</tr>
<tr>
<td>Gravity</td>
<td>9.81</td>
<td>( \frac{m}{s^2} )</td>
</tr>
<tr>
<td>Ixx</td>
<td>0.02</td>
<td>kg \cdot m^2</td>
</tr>
<tr>
<td>Iyy</td>
<td>0.02</td>
<td>kg \cdot m^2</td>
</tr>
<tr>
<td>Izz</td>
<td>0.04</td>
<td>kg \cdot m^2</td>
</tr>
</tbody>
</table>

Table 1. Model Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_d )</td>
<td>1.5</td>
</tr>
<tr>
<td>( K_d )</td>
<td>25</td>
</tr>
<tr>
<td>( K_p )</td>
<td>25</td>
</tr>
<tr>
<td>( K_d )</td>
<td>1.5</td>
</tr>
<tr>
<td>( K_p )</td>
<td>1.3</td>
</tr>
<tr>
<td>( K_p )</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 2. FOSMC Parameters

![Figure 2. 3d Plot of the Position](image-url)

The position tracking errors, this is \( e = \xi - \xi_d \), are shown in Figure 3. One can notice that they are close to zero. Also the position tracking errors for \( x \) and \( y \) raise to almost 4 centimetres when there is a transition from take-off to lemniscata and from lemniscata to landing. On the other hand the error in \( z \) decreases to 1 centimeter due to the disturbance that appeared at 10.9 seconds, but is close to zero in the rest of the trajectory.
Parameter | Value | Units
---|---|---
Max Thrust | 35 | N
Max Roll Torque | 4 | Nm
Max Pitch Torque | 4 | Nm
Max Yaw Torque | 2 | Nm

Table 3. Real Quadrotor Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\epsilon)</td>
<td>0.005</td>
</tr>
<tr>
<td>(\sigma_z)</td>
<td>0.1</td>
</tr>
<tr>
<td>(\lambda_z)</td>
<td>6.0</td>
</tr>
<tr>
<td>(\mu_z)</td>
<td>6.0</td>
</tr>
<tr>
<td>(\sigma_y)</td>
<td>0.1</td>
</tr>
<tr>
<td>(\lambda_y)</td>
<td>6.0</td>
</tr>
<tr>
<td>(\mu_y)</td>
<td>6.0</td>
</tr>
<tr>
<td>(\sigma_x)</td>
<td>0.1</td>
</tr>
<tr>
<td>(\lambda_x)</td>
<td>6.0</td>
</tr>
<tr>
<td>(\mu_x)</td>
<td>6.0</td>
</tr>
<tr>
<td>(K_{d_\phi})</td>
<td>1.5</td>
</tr>
<tr>
<td>(K_{p_\phi})</td>
<td>25</td>
</tr>
<tr>
<td>(K_{d_\psi})</td>
<td>1.5</td>
</tr>
<tr>
<td>(K_{p_\psi})</td>
<td>25</td>
</tr>
<tr>
<td>(K_{d_\theta})</td>
<td>1.3</td>
</tr>
<tr>
<td>(K_{p_\theta})</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 4. IOSMC Parameters

The control signals are shown in Figure 8 to Figure 11. It is important to mention that these control signals are saturated to the values shown in Table 3 [11], since in a real application the absolute value of those signals are limited due to the actuators (motors).

A parametric disturbance is introduced at 10.9 seconds. The disturbance consists of a change of the quadrotor’s nominal mass and inertias, the change is 6% in mass and 35% in inertias.

In order to compare the performance of the FOSMC design presented here, a IOSMC was integrated for the quadrotor’s simplified model. Using the small angle approximation discussed above this controller is given by

\[
P_z = \ddot{z}_d - \lambda_z \dot{e}_z - \sigma_z sgn(s_z) - \mu_z s_z,
\]

\[
P_y = \ddot{y}_d - \lambda_y \dot{e}_y - \sigma_y sgn(s_y) - \mu_y s_y,
\]

\[
P_x = \ddot{x}_d - \lambda_x \dot{e}_x - \sigma_x sgn(s_x) - \mu_x s_x.
\]

where as before, \(\lambda_z, \lambda_y, \lambda_x, \sigma_z, \sigma_y, \sigma_x, \mu_z, \mu_y\) and \(\mu_x\), are real coefficients chosen in such a way that the corresponding first order linear differential equations defined by the sliding surfaces

\[
s_z = 0 = \dot{e}_z + \lambda_z e_z,
\]

\[
s_y = 0 = \dot{e}_y + \lambda_y e_y,
\]

\[
s_x = 0 = \dot{e}_x + \lambda_x e_x.
\]

asymptotically converge to zero. The simulation carried out with this controller is the same used with the FOSMC using the control parameters shown in Table 4. The tracking position errors are shown in Figure 4 where one can be noticed that these errors are close to zero.
The errors in Euler Angles for the FOSMC are shown in Figure 5, these are close to zero. If we compare these errors with the errors in Euler Angles for the IOSMC shown in Figure 6, one can notice that the error in the yaw angle ($\psi$) is bigger in the IOSMC than in the FOSMC scheme.

A comparison between the trajectory tracking of the FOSMC and IOSMC scheme was carried out, we can conclude that they behave in a similar way but the FOSMC has a better yaw angle performance.

The errors in Euler Angles in both control schemes are around 0.4 [rad] (22.9°) except when there is a transition in the trajectory, this is when there is a change from take-off to the lemniscata trajectory or from the lemniscata trajectory to landing. Notice that the Euler Angles shown in Figure 7 are around 0.3 [rad] (17.2°), therefore the small angle approximation is valid.

![Figure 5. Errors in Euler Angles from FOSMC scheme](image)

![Figure 6. Errors in Euler Angles from IOSMC scheme](image)

![Figure 7. Euler Angles from FOSMC scheme](image)

![Figure 8. Control signal u1 from the FOSMC](image)

![Figure 9. Control signal u2 from the FOSMC](image)
The control signals for the FOSMC and the IOSMC are very similar, here we are only showing the FOSMC graphs, but if we compare the phase diagram of the yaw angle from the FOSMC and the IOSMC, depicted in Figure 13 and Figure 14 respectively, one can notice there is an extra loop in the IOSMC graph, that loop is causing a total increment of 0.01 radians in the IOSMC scheme. If we compare the graphs from the control input u4, depicted in Figure 11 and Figure 12, one can notice that the control input for the IOSMC scheme is higher than the FOSMC counterpart, one can notice that the only difference between both control schemes, FOSMC and IOSMC respectively is the yaw torque (control input u4), by inspecting the above mentioned graphs we conclude that the FOSMC has better yaw torque performance.

5 Experimental Platform

In order to test the control strategy proposed in this paper, experimental results are intended to being carried out. The quadrotor that will be used in the experiments is the one depicted in Figure 1. In order to obtain the states $\xi$ and $\eta$, two sensors are being used, one is an OptiTrack System that allows to have the position of the quadrotor
and an inertial measurement unit (IMU) which allows to have a measurement of the Euler Angles. In order to compute the fractional derivatives of these states a Grünwald-Letnikov fractional method is used with a fractional order of $\alpha = 0.9999$. The communication is a key problem for the experimental platform, therefore a Raspberry Pi3 with ROS installed will be used together with a VRPN library to obtain the position in a good approximation of a real-time system. The IMU will be also implemented with a library for Robotic Operating System, (ROS). The FOSMC strategy will be first implemented in hover then a trajectory in the space will be introduced.

6 Conclusion

This paper presents a Fractional Order Sliding Mode Control for the trajectory tracking of a Lemniscata trajectory. The model used is a simplified one, but it contains acceptable information for simulation results; this control scheme is based on previous results [1]. The simulation results obtained show good performance of the control scheme in the presence of disturbances. A disturbance in the quadrotor’s parameters, mainly the mass (above the 30% percent of its nominal value) is not dissipated by neither the FOSMC nor the IOSMC control scheme. The FOSMC has a very similar behaviour with the IOSMC, and has better performance in the yaw angle error, therefore both schemes are showing a good performance in the presence of uncertainties and disturbances.

Acknowledgements

The authors would like to thank Consejo Nacional de Ciencia y Tecnología CONACYT, Mexico, under grant 786275 and grant CB-2019-01.254329.

References


