CONTROL OF A QUADROTOR UNDER SENSORS SAMPLING LIMITATIONS USING TS FUZZY MODELING: A DISCRETE TIME APPROACH

J. G. Grijalva*, Michael Klug†, Eugênio B. Castelan*

*PPGEAS, DAS, UFSC, 88040-900, Florianópolis, SC, Brazil
†Federal Institute of Santa Catarina, IFSC, 89220-200, Joinville, SC, Brazil

Emails: gabriel.grijalva@posgrad.ufsc.br, michael.klug@ifsc.edu.br, eugenio.castelan@ufsc.br

Abstract—In this paper, we propose a procedure to obtain an algorithm, which can be embedded in any open source processor, based on nonlinear control for a quadrotor under sensors sampling time limitations using Takagi-Sugeno fuzzy modeling. Readings from the sensors can occur up to every limited time. The ultrasonic sensor and the inertial measurement unit work in low frequency compared with the other electronic devices involved. Thus, analysis in discrete time is developed along this work. Linear matrix inequalities theorems are used to design the feedback controller. A comparison with a continuous time approach is carried out to test the effectiveness of this controller.

Keywords—Robotics, control, discrete time, algorithm.

1 Introduction

Quadrotors are widely used nowadays in many applications such as in security and surveillance tasks, production of videos for different purposes, by general public like a hobby and by researchers to implement new technologies and ideas over these devices. From a control system perspective, a quadrotor is inherently a nonlinear and unstable dynamical system that needs some feedback controllers to carry through different tasks as hovering and tracking references, among others. Thus, many linear and nonlinear controllers have been proposed in the literature aiming at improving the stability and performance of quadrotors in different situations (Cisneros et al., 2016; Choi and Ahn, 2015; Lee and Kim, 2014).

Among the nonlinear techniques used for synthesizing the control laws, the so-called Parallel Distributed Compensation (PDC) (Wang et al., 1996) technique based on the use of TS fuzzy models has been shown to be attractive both from the practical and theoretical point of views. The PDC setting offers a simple and natural procedure to handle the nonlinear control systems with the possibility to engage the basic knowledge of linear systems and to use tools from the robust and linear parameter varying (LPV) control (Klug, 2015). For example, effective solutions in continuous-time to track references are presented in Yacef et al. (2012) and Lee and Kim (2014). However, to implement these control laws in digital processors, it is necessary to discretize them. Thus, if the discretization technique and/or the sampling time are not chosen correctly, the closed-loop performance can deteriorate, up to making the system unstable, or lead to excessive data processing. In special, the sampling rates of the ultrasonic sensor and the I.M.U. considered in the present work are critical points due to the capture speed they are featured with. Such kind of practical constraints and limitations should be considered to effectively implement a control law.

In Quadrotors applications, Remote Trajectory Generators (R.T.G) are used to emit the references to the plant. A personal computer is employed in Gautam and Ha (2013) to track positional references in X, Y and Z axis. On the other hand, numerous people such as cameramen, athletes and landscape video recording amateurs use a joystick as interface, and in this case pitch, roll, yaw angles and altitude are the states to be reached. Thus, a linear parameter varying (LPV) controller effectiveness has been tested in Cisneros et al. (2016) for high speed trajectory tracking. In Torres et al. (2016), it is presented an attitude controller in continuous-time considering the ultrasonic sensor and the I.M.U through eight local submodels. Small number of submodels and rules in fuzzy modeling causes a lighter processing than a large (Klug, 2015). Thus, to obtain less submodels without loosing information about the dynamics of the system we apply a simpler modeling, the sector nonlinearity method (Ohtake et al., 2003) based on weighting and membership functions. Also, in contrast to Yacef et al. (2012) and Lee and Kim (2014), we assume the employment of joystick without positional control. Thereby, this work is highlighted with the inclusion of these features in the control law design (limited sampling rates and use of a joystick interface) which is totally developed in discrete-time.

The paper is organized as follows. In section 2, we describe the considered control structure, which is based in a classical cascade of two loops that is used here to take into account the different sampling rates associated to the sensor and to the I.M.U, and present the nonlinear discrete-time model used to design the controllers. A brief review of the TS fuzzy modeling and an LMI condi-
tion to synthesize PDC controllers which guarantees some closed-loop performance are presented in section 3. The application of these techniques to the quadrotor problem is developed in section 4. Simulations and discussion of the results are presented in section 5. The paper ends with some concluding remarks.

2 Control-loop description

We dispose the structure of control in Figure 1.

![Figure 1: Control structure](image)

It is possible to identify two control loops. The control loop that corresponds to the attitude controller is in charge of the euler angles, operating with a sampling time \( T_a \) defined according to the bandwidth of the I.M.U. The same analysis is done with the altitude control loop, the ultrasonic sensor reading frequency determines the sampling time \( T_b \) to be used in. This control structure allows to work singly each control loop. In contrast to (Yacef et al., 2012), this control loop can set a dynamic in the attitude controller faster than the one of the altitude controller. It could be interpreted like an extra degree of freedom to design our control algorithm.

The controller is designed under the scheme depicted in Figure 2. The references \( \phi_r, \theta_r, \psi_r \) and \( Z_r \) are generated from the R.T.G which could be for example a joystick. The electronic devices on board are detailed as follows:

- **Modem:** Through wireless network, it receives the references from the R.T.G, next it sends this information to the processor.

- **Processor:** The algorithm developed is embedded in this part, it could be called the brain of the system because it is in charge of receive the information from the sensors, then it sends orders to the actuators.

- **I.M.U:** This micro electro-mechanical system provides to the processor information about the euler angles and its rates \((\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi})\).

- **Ultrasonic sensor:** The altitude of the quadrotor is measured by this element.

- **Electronic speed controller (E.S.C):** This part interacts between the processor and the actuators demodulating- conditioning the signals.

The critical rates between these devices are in the sensors. Accordingly, these parameters are considered as our starting point.

2.1 Quadrotor dynamic system

There are several works, see: (Mahony et al., 2012; Patel et al., 2012; Gaitan and Bolea, 2013), describing step by step the dynamic modeling of quadrotors. Therefore, we omit this procedure. The dynamic system used in this paper is a discrete time approximation based in Euler-forward method of the model presented in Yacef and Boudjema (2011). The choice of this model was done from the fact that it is valid for large angle variations. Also, it includes aerodynamic effects, such as air friction, making this model closer to the actual plant.

The states \(x, \dot{x}, \dot{\theta}, \psi, Z, \dot{Z}\) are represented by \(x_1, x_2, x_3, x_4, x_5, x_6, x_7\) and \(x_8\) respectively. Axis orientation \((X, Y, Z)\) of the plant and rotor speed \(\Omega_r\) of each motor are based in the scheme of Figure 2. Defining: \(x_a(k) = [x_1(k) x_2(k) x_3(k) x_4(k) x_5(k) x_6(k)]^T\) and \(U_a(k) = [U_2 U_3 U_4]^T\), the rotational subsystems is expressed by the following state equation:

\[
x_a(k+1) = \begin{bmatrix}
1 & T_a & 0 & 0 & 0 & 0 \\
0 & 1 & T_a & 0 & 0 & 0 \\
0 & 0 & 1 & T_a & 0 & 0 \\
0 & 0 & 0 & 1 & T_a & 0 \\
0 & 0 & 0 & 0 & 1 & T_a \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} x_a(k)
+ \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} U_a(k) + \begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix} \Omega_r(k) \tag{1}
\]

where:

\[
a_1 = \frac{I_2 - I_3}{I_x}, \quad a_2 = \frac{J_x}{I_n}, \quad a_3 = \frac{I_3 - I_4}{I_y}, \quad a_1 = \frac{J_r}{I_n}, \quad b_1 = \frac{L}{I_x}, \quad b_2 = \frac{L}{I_y}, \\
b_3 = \frac{L}{I_x} a_5 = \frac{I_5 - I_6}{I_x}, \quad e_1 = \frac{k_{m1}}{I_x}, \quad e_2 = \frac{k_{m2}}{I_y}, \quad e_3 = \frac{k_{m3}}{I_x}
\]
Next, defining \( x_b(k) = [x_T(k) \ x_S(k)]^T \) and \( e_6 = \frac{K_{rz}}{m_q} \), the translational subsystem that describes the altitude of the quadrotor is:

\[
x_b(k + 1) = \begin{bmatrix} 1 & T_b \ 0 & (1 + T_b e_6) \end{bmatrix} x_b(k) + \begin{bmatrix} 0 \\ T_b e_6 \end{bmatrix} e_6 + \begin{bmatrix} 0 \\ T_b g_r \end{bmatrix} \tag{2}
\]

Description of each symbol used in (1) and (2) is depicted in table 1:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_q )</td>
<td>Quadrotor mass</td>
</tr>
<tr>
<td>( q_r )</td>
<td>Gravity</td>
</tr>
<tr>
<td>( L )</td>
<td>Quadrotor axis length</td>
</tr>
<tr>
<td>( I_{xx}, I_{yy} ) and ( I_{zz} )</td>
<td>Inertias</td>
</tr>
<tr>
<td>( J_r )</td>
<td>Rotor inertia</td>
</tr>
<tr>
<td>( K_{a}, K_{ay}, K_{az} ) and ( K_{rz} )</td>
<td>Aerodynamic coeffs</td>
</tr>
<tr>
<td>( \Omega_r )</td>
<td>Rotor relative speed</td>
</tr>
<tr>
<td>( U_1, U_2, U_3 ) and ( U_4 )</td>
<td>Control inputs</td>
</tr>
</tbody>
</table>

Then, considering the thrust (\( K_f \)) and drag (\( K_m \)) coefficients, the speed of each rotor can be calculated with:

\[
\begin{bmatrix}
\Omega_1^2 \\
\Omega_2^2 \\
\Omega_3^2 \\
\Omega_4^2 \\
\end{bmatrix} = \begin{bmatrix}
\frac{1}{4} \ \mathbf{1} & 0 & 0 & 0 \\
\frac{1}{4} & \frac{1}{2} \ \mathbf{1} & 0 & 0 \\
\frac{1}{4} & 0 & \frac{1}{2} \ \mathbf{1} & 0 \\
\frac{1}{4} & 0 & 0 & \frac{1}{2} \ \mathbf{1} \\
\end{bmatrix} \begin{bmatrix}
\Omega_1 \\
\Omega_2 \\
\Omega_3 \\
\Omega_4 \\
\end{bmatrix} + \begin{bmatrix}
\mathbf{1} \\
\mathbf{1} \\
\mathbf{1} \\
\mathbf{1} \\
\end{bmatrix} (\Omega_1 \ x_{1 \omega} + \Omega_2 \ x_{2 \omega} + \Omega_3 \ x_{3 \omega} + \Omega_4 \ x_{4 \omega})
\tag{3}
\]

### 3 TS fuzzy approach

In this section we present a brief summary of the TS fuzzy modeling and the LMIs stabilization theorem used to develop our controller.

#### 3.1 Modeling

Let represent a nonlinear plant by:

\[
x_{k+1} = A(x_k)x_k + B(x_k)u_k \\
y_k = Cx_k
\tag{4}
\]

with \( x_k \in \mathcal{X} \subseteq \mathbb{R}^{n_x} \), \( u_k \in \mathcal{U} \subseteq \mathbb{R}^{n_u} \) and \( y_k \in \mathcal{Y} \subseteq \mathbb{R}^{n_y} \). The functions \( A(\cdot) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x} \) and \( B(\cdot) : \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x \times n_u} \) are continuous and bounded for all \( x \in \mathcal{X} \). \( \mathcal{X} \) is defined as a region that belongs to the state space domain containing the origin and \( C \in \mathbb{R}^{n_y \times n_x} \) is a matrix with constant values. The TS model for (4) is:

\[
R_i: \begin{cases}
\text{IF} & z_{i(1)} \text{ is } M^i_{1}, \ z_{i(2)} \text{ is } M^i_{2}, \ldots, \ z_{i(n_x)} \text{ is } M^i_{n_x} \\
\text{THEN} & x_{k+1} = A_i x_k + B_i u_k \\
y_k = C_i x_k
\end{cases}
\]

where \( R_1, \ldots, R_{n_r} \) are fuzzy rules, \( z_k := [z_k(1), z_k(2), \ldots, z_k(n_x)] \) represent the premise variables, \( M^i_j, j = 1, \ldots, n_x \), represent the fuzzy sets and \( A_i, B_i \) are the matrices that define the fuzzy local submodels. Now, let consider \( h_j^i(z_{k(j)}) \) as the weights of the fuzzy sets \( M^i_j \) associated to the premise variable \( z_{k(j)} \), and \( \omega^i(z) = \prod_{j=1}^{n_x} h_j^i (z_{k(j)}) \).

Assuming \( h_j^i (z_{k(j)}) \geq 0 \), it implies \( \omega^i(z_k) \geq 0 \), \( \forall i = 1, \ldots, n_r \) and \( \sum_{i=1}^{n_r} \omega^i(z_k) > 0 \). Thus, we have the TS fuzzy model in discrete time:

\[
x_{k+1} = A(h_k)x_k + B(h_k)u_k
\tag{5}
\]

where:

\[
[A(h_k) \ B(h_k)] = \sum_{i=1}^{n_r} h_{ki} [A_i \ B_i]
\]

Notice that the fuzzy model (5) is equivalent to a LPV system.

#### 3.2 Stabilization

In PDC design, each control rule is defined from the corresponding rule of a TS fuzzy model:

\[
CR_i: \begin{cases}
1F & z_{i(1)} \text{ is } M^i_{1}, \ z_{i(2)} \text{ is } M^i_{2}, \ldots, \ z_{i(n_x)} \text{ is } M^i_{n_x} \\
\text{THEN} & u_k = K_i x_k
\end{cases}
\tag{6}
\]

In (6) \( CR_i \) represents each fuzzy control rule. This have a linear controller in the consequent parts. The overall fuzzy controller is represented by:

\[
u_k = K(h_k)x_k \tag{7}
\]

with:

\[
[K(h_k)] = \sum_{i=1}^{n_r} h_{ki} [K_i]
\]

**Theorem 1** (Klug et al., 2011). Given a contravariant coefficient \( \lambda \in (0,1) \). If there exist definite positive matrices \( Q_i \in \mathbb{R}^{n_x \times n_x} \) and matrices \( U \in \mathbb{R}^{n_u \times n_x}, Y_i \in \mathbb{R}^{n_u \times n_x} \) satisfying the condition in (8):

\[
\begin{bmatrix}
Q_i \\
Y_i \\
\end{bmatrix} \begin{bmatrix}
(A_i + A_j)U + B_i Y_i + B_j Y_i \\
2 \lambda^{-1} \left( Q_i + Q_j \right) - U^T U
\end{bmatrix} < 0
\tag{8}
\]

\[\forall q, i = 1, \ldots, n_r \text{ and } \forall j = i, \ldots, n_r.\]

Then, the closed loop system \( (A(h_k) + B(h_k)K(h_k))x_k \) where \( A(h_k) \) and \( B(h_k) \) are known matrices, with gains in (7) given by \( K_i = Y_i U^{-1} \), is asymptotically stable for any initial condition \( x_0 \in \mathbb{R}^{n_x} \).
Firstly, let define the premise variables

\[
\begin{align*}
4.1 \text{ Rotational subsystem} \\
4.2 \text{ Translational subsystem}
\end{align*}
\]

\[
\begin{align*}
\text{Figure 3 illustrates with dashed lines the region in which theorem 1 locates the poles through the relation } \lambda = e^{-\alpha T_{s}}, \text{ being } -\alpha \text{ the position on the real axis in the S-plane and } T_{s} \text{ the sampling time. More detailed information about this theorem can be found in Klug et al. (2011).}
\end{align*}
\]

4 Application

The approach showed in the last section is applied in the subsystems of our plant:

4.1 Rotational subsystem

4.1.1 TS fuzzy modeling.

Firstly, let define the premise variables \(z_1 = x_2(k)\) and \(z_2 = x_4(k)\). We consider the maximum rates of \(\phi\) and \(\theta\) in our design equal to \(M_{RP}\) and \(M_{RT}\) respectively. Because of the symmetry of the quadratric the minimum values are expressed as the negative maximum values. Now, we calculate the maximum and minimum values of \(z_1\) and \(z_2\) under \(x_2(k) \in [-M_{RP}, M_{RP}]\), and \(x_4(k) \in [-M_{RT}, M_{RT}]\) as follows:

\[
\begin{align*}
\text{max}_{x_2, x_4} z_1(k) &= M_{RP} = q_1, \quad \text{min}_{x_2, x_4} z_1(k) = -M_{RP} = q_2 \\
\text{max}_{x_2, x_4} z_2(k) &= M_{RT} = f_1, \quad \text{min}_{x_2, x_4} z_2(k) = -M_{RT} = f_2
\end{align*}
\]

The membership functions can be calculated as:

\[
\begin{align*}
V_1(z_1) &= \frac{z_1(k) - q_2}{q_1 - q_2}, \\
V_2(z_1) &= \frac{q_1 - z_1(k)}{q_1 - q_2}, \\
W_1(z_2) &= \frac{z_2(k) - f_2}{f_1 - f_2}, \\
W_2(z_2) &= \frac{f_1 - z_2(k)}{f_1 - f_2}
\end{align*}
\]

From (1), we obtain the fuzzy model by using (9) and (10):

\[
x_{a}(k+1) = \sum_{i=1}^{3} \sum_{j=1}^{3} V_i(z_1(k))W_j(z_2(k))
\]

\[
\begin{bmatrix}
\begin{array}{cccc}
0 & 0 & 0 & 0 \\
1 & T_a & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{array}
\end{bmatrix}
\begin{bmatrix}
0 \\
b_1 T_a \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
+ \begin{bmatrix}
0 \\
-\alpha_2 f_1 T_a \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
\begin{bmatrix}
\Omega_r(k) \\
\end{bmatrix}
\]

The defuzzification of (11) is carried out as:

\[
x_{a}(k+1) = \sum_{i=1}^{4} h_i(z_1(k)) (A_{ai} x_a(k) + B_{ai} U_a(k) + G_{ai} \Omega_r(k))
\]

where:

\[
\begin{align*}
h_1(z_1) &= V_1(z_1) \times W_1(z_2) \\
h_2(z_1) &= V_1(z_1) \times W_2(z_2) \\
h_3(z_1) &= V_2(z_1) \times W_1(z_2) \\
h_4(z_1) &= V_2(z_1) \times W_2(z_2)
\end{align*}
\]

Notice that \(i = j + 2(t - 1)\). The fuzzy model (11) represents exactly the rotational equations of motion of the rotator in the region \([-M_{RP}, M_{RP}] \times [-M_{RT}, M_{RT}]\) on the \(x_2(k)\) and \(x_4(k)\) space respectively under convexity conditions.

4.1.2 Control.

The attitude controller is defined as follows:

\[
U_a(h) = [K_a(h)x_a(k) + \Gamma(h)\Omega_r(k)]
\]

with:

\[
\Gamma(h) = \begin{bmatrix}
\alpha_2 f_1(h) \\
\alpha_2 f_1(h) \\
\alpha_2 f_1(h) \\
\alpha_2 f_1(h) \\
\alpha_2 f_1(h) \\
\alpha_2 f_1(h) \\
\alpha_2 f_1(h) \\
\alpha_2 f_1(h) \\
\alpha_2 f_1(h) \\
\end{bmatrix}
\]

In (12), \(\Gamma(h)\) compensates dynamically the rotator relative speed \(\Omega_r\). Thereby, in closed loop we have:

\[
x_{a}(k+1) = [A_a(h(k)) + B_a K_a(h(k))] x_a(k)
\]

Considering (11) and (12), theorem 1 can be used to compute the fuzzy gains \(K_a(h(k))\) such that (13) is robust asymptotically stable.

4.2 Translational subsystem

4.2.1 TS fuzzy modeling.

The maximum angle of slope under the fuzzy model exactly represents the dynamic of the plant is \(M_A\). The modeling is done under \(x_3(k) \in [M_A, -M_A]\). We define the premise \(z_3 = \cos x_3(k) \cos x_3(k)\). After, we calculate the minimum and maximum values of \(z_3\):

\[
\begin{align*}
\text{max}_{\cos x_3, \cos x_3} z_3(k) &= 1 = d_1, \quad \text{min}_{\cos x_3, \cos x_3} z_3(k) = \cos^2(M_A) = d_2
\end{align*}
\]

With the next membership function:

\[
\begin{align*}
J_1 &= \frac{z_3(k) - d_2}{d_1 - d_2}, \\
J_2 &= \frac{d_1 - z_3(k)}{d_1 - d_2}
\end{align*}
\]
Employing (14) and (15) the fuzzy model of (2) is:

\[
x_b(k + 1) = \sum_{i=1}^{2} J_i \left\{ \frac{A_i x_b + B_i U_1 + G}{\lambda_i} \right\} + \left[ \frac{\tau_{dbk}}{m_c} \right] x_b(k) + \left[ \frac{\tau_{dsk}}{c} \right] U_1(k) = \sum_{i=1}^{2} \overline{P}_i(z) x_b(k) + \left[ \frac{\tau_{dbk}}{m_c} \right] x_b(k) + \left[ \frac{\tau_{dsk}}{c} \right] U_1(k)
\]

(16)

The defuzzification of (16) is carried out as:

\[
x_b(k + 1) = \sum_{i=1}^{2} \overline{P}_i(z) x_b(k) + \left[ \frac{\tau_{dbk}}{m_c} \right] x_b(k) + \left[ \frac{\tau_{dsk}}{c} \right] U_1(k)
\]

where:

\[
\overline{P}_1(z(k)) = J_1(z(k)) \quad \overline{P}_2(z(k)) = J_2(z(k))
\]

The fuzzy model (16) represents exactly the translational equations of motion of the quadrotor in the region \([-MA, MA]\) on the z_3 space under convexity conditions.

### 4.2.2 Control

To avoid steady state error, we add an integral action: \(\xi(k) = \xi(k + 1) - y(k)\) in the PDC. Accordingly, the control law is defined as:

\[
U_1(k) = [K_b(\overline{P}) x_b(k) - F(\overline{P}) \xi(k) - m_q g_r]
\]

(17)

So, the augmented altitude system in closed loop is:

\[
\overline{x}_b(k + 1) = [\overline{A}_b + \overline{B}_b(\overline{P}(k)) K(\overline{P}(k))] \overline{x}_b(k)
\]

(18)

with:

\[
\overline{x}_b = \begin{bmatrix} x_b \\ \xi \end{bmatrix}, \overline{A}_b = \begin{bmatrix} A_b & 0 \\ C & I \end{bmatrix}, \overline{B}_b = \begin{bmatrix} B_b \\ 0 \end{bmatrix}, K = [K_b \ -F]
\]

Now, the system (18) provides zero steady state error. Moreover, the system is robust to uncertainties in the model of the plant and rejects step disturbances.

### 4.3 Remark 1

Taking advantage of the analysis done in this section, we propose a generic algorithm for quadrotors which guarantees stability, under the conditions predefined, and efficient processing. To complement Algorithm 1, it is possible to use (3) to compute the commanded rotor speed corresponding to the inputs of the E.S.C of each DC motor.

As the calculations of the gains \(K_a\), \(K_b\) and \(F\) through LMIs are performed offline, this algorithm can be easily embedded in any open source processor.

### Algorithm 1 Generic Algorithm Structure

1. procedure WEIGHT GEN.(Attitude Control)
2. for \(t = 1 \to 2\) do
3. for \(j = 1 \to 2\) do
4. \(\ell = j + 2(t - 1)\);
5. \(aux_{1[j]} = V_{[j]} W_{[j]}\); \(\triangleright\) see (10)
6. \(aux_{2} = 0\);
7. for \(\ell = 1 \to 4\) do
8. \(aux_{2} = aux_{2} + aux_{1[j]}\);
9. \(iaux_{2} = 1/aux_{2}\);
10. for \(\ell = 1 \to 4\) do
11. \(h_{[j]} = aux_{1[j]}iaux_{2}\);
12. procedure CONTROL LAW(Attitude Control)
13. \(U_a = 0\);
14. for \(i = 1 \to 4\) do
15. \(aux_{3[i]} = h_{[i]} K_a[i] F_{a}\); \(\triangleright\) see (12)
16. \(aux_{4[i]} = h_{[i]} \Gamma_{[i]} \Omega_{r}\);
17. \(U_a = U_a + aux_{3[i]} + aux_{4[i]}\);
18. procedure WEIGHT GEN.(Altitude Control)
19. \(h_{[1]} = J_{[1]}\); \(h_{[2]} = J_{[2]}\); \(\triangleright\) see (15)
20. procedure CONTROL LAW(Altitude Control)
21. \(U_1 = h_{[1]} K_b[1] x_b + h_{[2]} K_b[2] x_b - h_{[1]} F_{[1]} \xi - h_{[2]} F_{[2]} \xi - m_q g_r\); \(\triangleright\) see (17)

### 5 Simulation results and discussion

The values of the parameters used in this simulation taken from Yacef and Boudjema (2011) are depicted in table 2.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Values and units</th>
<th>Symbol</th>
<th>Values and units</th>
</tr>
</thead>
<tbody>
<tr>
<td>(m_b)</td>
<td>0.486 Kg</td>
<td>(L)</td>
<td>0.255 m</td>
</tr>
<tr>
<td>(g_r)</td>
<td>9.81 m/s^2</td>
<td>(J_r)</td>
<td>3.35 \times 10^{-5} Kg.m^2</td>
</tr>
<tr>
<td>(I_{xx})</td>
<td>3.82 \times 10^{-5} Kg.m^2</td>
<td>(I_{yy})</td>
<td>3.82 \times 10^{-5} Kg.m^2</td>
</tr>
<tr>
<td>(I_{zz})</td>
<td>7.65 \times 10^{-4} Kg.m^2</td>
<td>(K_{u})</td>
<td>5.567 \times 10^{-3}</td>
</tr>
<tr>
<td>(K_w)</td>
<td>5.567 \times 10^{-4}</td>
<td>(K_{e})</td>
<td>6.354 \times 10^{-4}</td>
</tr>
<tr>
<td>(K_{e})</td>
<td>0.048</td>
<td>(K_{m})</td>
<td>1.2 \times 10^{-6}</td>
</tr>
<tr>
<td>(K_{f})</td>
<td>2.932 \times 10^{-5}</td>
<td>(M_A)</td>
<td>(\pi/3) rad</td>
</tr>
<tr>
<td>(M_{RF})</td>
<td>5 rad/s</td>
<td>(M_{RF})</td>
<td>5 rad/s</td>
</tr>
<tr>
<td>(T_a)</td>
<td>0.05 s</td>
<td>(T_b)</td>
<td>0.1 s</td>
</tr>
</tbody>
</table>

The sampling times \(T_a\) and \(T_b\) were chosen according to the average of bandwidths in commercial I.M.U and ultrasonic sensors. It is assumed the use of brushless DC motors that provide high torque and little friction. The parameters that describe the features of these motors are the gain and the time constant. The transfer function maps the desired propeller speed to the actual speed. The voltage supplied to the motors is directly proportional to the \([\text{rad/s}]\) of its rotation. The constant of proportionality of this linear relationship appears as a gain in the transfer function (Cisneros et al., 2016) in (19).

\[
\Lambda(s) = \frac{\text{Actual rotor speed}}{\text{Commanded rotor speed}} = \frac{0.936}{0.178s + 1}
\]

(19)
Looking for a response in the attitude controller faster than the altitude controller we assign the values of $\lambda$ in the LMI solver developed by Gahinet et al. (1994), which allocates the poles for discrete time approach in 0.9 and 0.6 for the altitude and attitude controller, respectively.

Mostly, quadrotors joysticks have configurable buttons. The buttons (roll, pitch and yaw) are set with steps amplitude equal to $\pm(0.8, 0.8$ and 1$) rad$, respectively. These values, far from the equilibrium point of a linearized system, are useful to test the performance, such as high speed trajectory tracking, of the nonlinear controller. The plant starts over the ground it means 0 m, then it is demanded to hover in $-4$ m along the $Z$ axis.

Figure 4 shows the tracking performance: in blue solid line is presented the approach explained in this paper, black dashed line shows the behavior of a quadrotor with a TS fuzzy controller in continuous time without considering the sensors limitations.

The control inputs (moment in N.m.) are depicted in Figure 5, there we can see $U_1$ greater than the other inputs. $U_1$ is in charge of the altitude control which request to the actuators to increase plenty their work. The other inputs $U_2$, $U_3$ and $U_4$ modify the slope and the heading of the quadrotor, so it is requested only a small variation of the angular speed of the propellers. The minimum speed required to hover is $250 \text{ rad/s} \approx (2400 \text{ RPM})$, this can be inferred from Figure 6. The maximum value reached by the motors is under $400 \text{ rad/s} \approx (3800 \text{ RPM})$. Nowadays there is not problem to get brushless motors for quadrotor with top speed of (10000 RPM). Thus, we conclude that the controller is not demanding impossible actions to the actuators.

The parameter-varying characteristic of the TS fuzzy model based controllers with PDC used in this work can be observed in Figure 7 that shows how the weights fluctuate according to the rules predefined in the modeling for each premise variable.
6 Conclusion

The number of rules we used in the fuzzy modeling is smaller than the ones of the references cited in this paper. Notwithstanding we were developing a nonlinear controller, linear control techniques such as integral action with augmented dynamic were applied successfully. Furthermore the efficient processing advantages of this approach, the tracking performance is as satisfactory as the one offered in continuous time. Looking for save energy in batteries, the algorithm 1 is going to be tested versus other nonlinear algorithms to evaluate processing efficiency in quadrotors.

Acknowledgments

This research was supported by the Ecuadorian Council of research and technology SENESCYT and CAPES-Brazil.

References


