HAL
open science

Controlling a drone: Comparison between a based model method and a fuzzy inference system

Kadda Zemalache, Hichem Maaref

To cite this version:

Kadda Zemalache, Hichem Maaref. Controlling a drone: Comparison between a based model method and a fuzzy inference system. Applied Soft Computing, 2009, 9 (2), pp.553-562. 10.1016/j.asoc.2008.08.007 . hal-00345354

HAL Id: hal-00345354

https://hal.science/hal-00345354

Submitted on 11 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Controlling a drone: Comparison between a based model method and a fuzzy inference system

K.M. Zemalache *, H. Maaref

IBISC Laboratory, CNRS-FRE 3190, Université d'Evry Val d'Essonne, 40 Rue du Pelvoux, 91020 Evry Cedex, France

The work describes an automatically on-line self-tunable fuzzy inference system (STFIS) of a new configuration of mini-flying called XSF (X4 Stationnary Flyer) drone. A fuzzy controller based on on-line optimization of a *zero* order *Takagi–Sugeno* fuzzy inference system (FIS) by a back propagation-like algorithm is successfully applied. It is used to minimize a cost function that is made up of a quadratic error term and a weight decay term that prevents an excessive growth of parameters. Thus, we carried out control for the continuation of simple trajectories such as the follow-up of straight lines, and complex (half circle, corner, and helicoidal) by using the STFIS technique. This permits to prove the effectiveness of the proposed control law. Simulation results and a comparison with a static feedback linearization controller (SFL) are presented and discussed. We studied the robustness of the two controllers used in the presence of disturbances. We presented two types of disturbances, the case of a breakdown of an engine as well as a gust of wind.

1. Introduction

The past few years have witnessed a rapid growth in the use of fuzzy logic controllers for the control of processes, which are complex and ill defined. Most fuzzy controllers developed till now have been of the rule-based type, where the rules in the controller attempt to model the operator's response to particular process situation.

Recently, the resurgence of interest in the field of artificial neural networks has injected a new driving force into the *fuzzy* literature. The back-propagation learning rule, which drew little attention till its applications to artificial neural networks was discovered, is actually an universal learning paradigm for any smooth parameterized models, including fuzzy inference systems. As a result, a fuzzy inference system can now not only take linguistic information from human experts, but also adapt itself using numerical data (input/output pairs) to achieve better performance. This gives fuzzy inference systems an edge over neural networks, which cannot take linguistic information directly.

In autonomous wheeled robot, many developed learning techniques have arisen in order to generate or to tune fuzzy rules. Most of them are based on the so-called "neuro-fuzzy learning algorithms" as proposed by [22,23,25,26,32,39]. These methods are well for constructing an optimal fuzzy system model which is used to identify the corresponding practical system.

Modeling and controlling aerial vehicles (blimps and mini rotorcraft) are the principal preoccupation of the IBISC-group. Within this optic, who attracted the contest of the DGA-ONERA[1] was the XSF project which consists of a drone with revolving aerofoils [4](see Fig. 1 left). It is equipped with four rotors where two are directionals, what we call in the following X4 Stationary Flyer (XSF).

The XSF is an engine of 68 cm $\times$ 68 cm of total size and not exceed 2 kg in mass. It is designed in a cross form and made of carbon fiber. Each tip of the cross has a rotor including an electric brushless motor, a speed controller and a two-blade ducted propeller. The operating principle of the XSF can be presented thus: two rotors turn clockwise, and the two other rotors turn counterclockwise to maintain the total equilibrium in yaw motion. The equilibrium of angular velocities of all rotors done, the unmanned aerial vehicle (UAV) is either in stationary position, or moving vertically. A characteristic of the XSF compared to the

* Corresponding author.
E-mail addresses: zemalache@iup.univ-evry.fr, kadda.zemalache@ibisc.univ-evry.fr (K.M. Zemalache), hichem.maaref@ibisc.univ-evry.fr (H. Maaref).

[1] This work is supported by the mini-flyer competition program organized by the DGA (Direction Générale des Armements) and the ONERA (Office Nationale d'Etude et de Recherche en Aérospatiale), France.

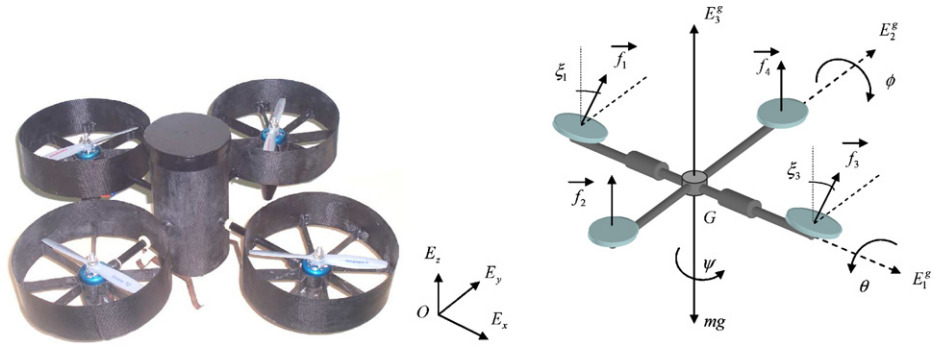

Fig. 1. Conceptual form of the XSF (left) and frames attached to the XSF (right).

existing quadrotors, is the swiveling of the supports of the motors 1 and 3 around the pitching axis thanks to two small servomotors controlling the swiveling angles ξ_1 and ξ_3. This swiveling ensures either the horizontal rectilinear motion or the rotational movement around the yaw axis or a combination of these two movements which gives the turn (see Fig. 1 right). This permits a more stabilized horizontal flight and a suitable cornering.

Drone XSF is a flying machine of low dimension able to carry a small payload, in particular a camera, intended to carry out in an autonomous way a reconnaissance mission. In addition to the military applications, this type of machines can also interest the civil field, in particular the search for anybody in buildings in flame, the environmental protection, the natural risk management, the management of the great infrastructures.

Several recent works were completed for the design and control in pilotless aerial vehicles domain such that Quadrotor [1,2,10,29,33,41,42]. Also, related models for controlling the vertical take-off and landing (VTOL) aircraft are studied by Hauser et al. [18]. A model for the dynamic and configuration stabilization of quasi-stationary flight conditions of a four rotors VTOL, based on Newton formalism, was studied by Hamel et al. [16] where the dynamic motor effects are incorporated and a bound of perturbing errors was obtained for the coupled system. Castillo et al. [13] performed autonomous take-off, hovering and landing control of a four rotors by synthesizing a controller using the Lagrangian model based on the Lyapunov analysis.

The stabilization problem of a four rotors rotorcraft is also studied and tested by Castillo [12] where the nested saturation algorithm is used, the input/output linearization procedure [18], in [11] a proportional integral derivative (PID) controller and a linear quadratic (LQ) controller were implemented and proved capable of regulating the system and application of the theory of flat systems by Beji and Abichou [6]. Mistler et al. [31] developed a dynamic model in non-linear state space representation, and used an exact linearization and non-interacting control for the global system to evaluate translational motion and yaw angle outputs. Mokhtari et al. [30] proposed an attempt to apply linear H_∞ outer control of helicopter quadrotor with plant uncertainty combined with a robust feedback linearization inner controller. Hanford et al. [19], presented a simple closed loop equipped with MEMS (micro-electro-mechanical systems) sensors and PIC based processing unit. Waslander et al. [40] maked an emphasis on the insufficiency of classical control methods and proposes the integral sliding mode controller associated with reinforcement learning to achieve multi agent control. Tayebi and McGilvray [37] proposed a new quaternion-based feedback control scheme for exponential attitude stabilization of a quadrotor. The proposed controller is based upon the compensation of the Coriolis and gyroscopic torques and the use of a PD^2 feedback structure, where the proportional action is in terms of the vector quaternion and the two derivative actions are in terms of the airframe angular velocity and the vector quaternion velocity. Hoffman et al. [20,21], achieved the formation control by sliding mode technique and focused on collision and obstacle avoidance by extracting the state variables with a Kalman filter. Mederreg et al. [28] developed a non-linear controller with observers based on the backstepping. The robustness of this controller is studied, performances and stability of the suggested controller are analyzed through simulations carried out on the model (kinematics and dynamic equations). In Benzemrane et al. [5], addressed the classical problem of speed estimation of an Unmanned Aerial Vehicle when the acceleration, the angles and the angular speeds are available for measurement. A solution has been provided for a class of systems via the tools of adaptive observation theory with promissing results. Bestaoui and Slim [8] addressed the problem of characterizing maneuvers paths on the group of rigid body motions in 3D for a quadrotor. The role of the trajectory generator is to generate a feasible time trajectory for the UAV.

Flight control methods utilizing vision systems are also studied by [38], which exploits the Moir patterns, and in [3], which utilizes double cameras. Hamel and Mahony [17] proposed a vision based controller which performs visual servo control by positioning a camera onto a fixed target for the hovering of a quadrotor.

All the reviewed techniques require the well knowledge of the system dynamic model and parameters. In this paper, a STFIS control strategy is developed based on the systems output measures is implemented. This technique early used for autonomous wheeled robot, is adapted and modified for the used with the XSF.

The arrangement of this paper is as follows. The dynamic model of the XSF drone is given in the second section. The developed ideas of control for the XSF by the self-tunable fuzzy inference system (STFIS) controller is presented and compared with a static feedback linearization controller (SFL) to stabilize the XSF by using the point to point steering stabilization in Section 3. Motion planning and simulation results are introduced in Section 4. The robustness of the proposed controller is then evaluated in Section 5. Finally, conclusion and future works are given in the last section.

2. Configuration description and modeling

Unlike regular helicopters that have variable pitch angles, an engine has fixed pitch angle rotors and the rotor speeds are controlled to produce the desired lift forces. Basic motions of the XSF can described using Fig. 1(right). Vertical motion is controlled by collectively increasing or decreasing the power for all motors. Lateral motion, in x direction or in y direction, is not achieved by differentially controlling the motors generating a pitching/rolling motion of the airframe that inclines the collective thrust (producing horizontal forces) and leads to lateral accelerations

[1,12,16]. But, two engines of direction are used to permute between the x and y motion.

The XSF is a system consisting of four individual electrical fans attached to a rigid cross frame. We consider a local reference airframe $\Re_G = \{G, E_1^g, E_2^g, E_3^g\}$ at G (mass center) while the inertial frame is denoted by $\Re_O = \{O, E_x, E_y, E_z\}$ such that the vertical direction E_z axis is pointing upwards. Let the vector $X = (x, y, z)$ denotes the G position with respect to $\Re_O$. The rotation of the rigid body is defined by $R_{\phi,\theta,\psi}$: $\Re_O \to \Re_G$, where $R_{\phi,\theta,\psi} \in \mathrm{SO}(3)$ is an orthogonal rotation matrix which is defined by the Euler angles, θ (pitch), ϕ (roll) and ψ (yaw), regrouped in $\eta = (\phi, \theta, \psi)$. An Euler angle representation given in Eq. (1) has been chosen [43].

$$R = \begin{pmatrix} C_\psi C_\theta & C_\theta S_\psi & -S_\theta \\ S_\phi C_\psi S_\theta - S_\psi C_\phi & S_\theta S_\psi S_\phi + C_\psi C_\phi & C_\theta S_\phi \\ S_\theta C_\psi C_\phi + S_\psi S_\phi & C_\phi S_\theta S_\psi - C_\psi S_\phi & C_\theta C_\phi \end{pmatrix} \tag{1}$$

where for example C_θ and S_θ represent $\cos\theta$ and $\sin\theta$, respectively.

The main feature of the presented XSF in comparison with the existing quadrirotors [1,12,16], is the swiveling of the actuators supports 1 and 3 around the axis of pitching (angles ξ_1 and ξ_3). This swiveling ensures either the horizontal rectilinear motion or the rotational movement around the yaw axis or a combination of these two movements which gives the turn (see Fig. 3(right)), as well as the direction of rotation of the rotors implies that rotors 1 and 2 turn clockwise and rotors 3 and 4 turn counterclockwise.

2.1. Motion dynamic

We consider the translation motion of $\Re_G$ with respect to $\Re_O$. The position of the mass center $wrt\Re_O$ is defined by $\overline{OG} = (x\,y\,z)^{\mathrm{T}}$, its time derivative gives the velocity wrt to $\Re_O$ such that $(\mathrm{d}\overline{OG})/(\mathrm{d}t) = (\dot{x}\dot{y}\dot{z})^{\mathrm{T}}$, while the second time derivative permits to get the acceleration $(\mathrm{d}^2\overline{OG})/(\mathrm{d}t^2) = (\ddot{x}\ddot{y}\ddot{z})^{\mathrm{T}}$ (Fig. 1(right)).

The model is a simplified one's. The constraints as external perturbation and the gyroscopic torques are neglected. The aim is to control the engine vertically z axis and horizontally according to x and y axis. The vehicle dynamics, represented in Fig. 1(right), is modeled by the system of following [4,7,45]:

$$\begin{aligned} m\ddot{x} &= S_\psi C_\theta u_2 - S_\theta u_3 \\ m\ddot{y} &= (S_\theta S_\psi S_\phi + C_\psi C_\phi)u_2 + C_\theta S_\phi u_3 \\ m\ddot{z} &= (S_\theta S_\psi C_\phi - C_\psi C_\phi)u_2 + C_\theta C_\phi u_3 - mg \end{aligned} \tag{2}$$

m is the total mass of the vehicle. The vector u_2 and u_3 combines the principal non-conservative forces applied to the engine airframe including forces generated by the motors and drag terms. Drag forces and gyroscopic due to motors effects will be not considered in this work. The lift (collective) force u_3 and the direction input u_2 are such that

$$\begin{pmatrix} 0 \\ u_2 \\ u_3 \end{pmatrix} = f_1 \acute{e}_1 + f_2 e_2 + f_3 \acute{e}_3 + f_4 e_4 \tag{3}$$

with $f_i = K_T \omega_i^2$ where $K_T = 10^{-5}$ N s^2 and ω_i is the angular speed resulting of motor i. Let

$$\acute{e}_1 = \begin{pmatrix} 0 \\ S_{\xi_1} \\ C_{\xi_1} \end{pmatrix}_{\Re_G}; \qquad \acute{e}_3 = \begin{pmatrix} 0 \\ S_{\xi_3} \\ C_{\xi_3} \end{pmatrix}_{\Re_G}; \qquad e_2 = e_4 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}_{\Re_G} \tag{4}$$

Then we deduce:

$$\begin{aligned} u_2 &= f_1 S_{\xi_1} + f_3 S_{\xi_3} \\ u_3 &= f_1 C_{\xi_1} + f_3 C_{\xi_3} + f_2 + f_4 \end{aligned} \tag{5}$$

ξ_1 and ξ_3 are the two internal degree of freedom of rotors 1 and 3, respectively. These variables are controlled by dc-motors and bounded $-20^\circ \le \xi_1, \xi_3 \le +20^\circ$. e_2 and e_4 are the unit vectors along E_3^g which imply that rotors 2 and 3 are identical of that of a classical quadrotor [43,44].

2.2. Rotational motion

The rotational motion of the XSF will be defined wrt to the local frame but expressed in the inertial frame.

Where the inertia elements I_{xx}, I_{yy} and I_{zz} are of the inertia matrix I_G expressed in G, then $I_G = \mathrm{diag}(I_{xx}, I_{yy}, I_{zz})$.

$$\begin{aligned} \ddot{\theta} &= \frac{1}{I_{xx} C_\phi}(\tau_\theta + I_{xx} S_\phi \dot{\phi}\dot{\theta}) \\ \ddot{\phi} &= \frac{1}{I_{yy} C_\theta C_\phi}(\tau_\phi + I_{yy} S_\phi C_\theta \phi^2 + I_{yy} S_\theta C_\phi \dot{\theta}\dot{\phi}) \\ \ddot{\psi} &= \frac{\tau_\psi}{I_{zz}} \end{aligned} \tag{6}$$

With the three inputs in torque

$$\begin{aligned} \tau_\theta &= l(f_2 - f_4) \\ \tau_\phi &= l(f_1 C_{\xi_1} - f_3 C_{\xi_3}) \\ \tau_\psi &= l(f_1 S_{\xi_1} - f_3 S_{\xi_3}) + \frac{K_M}{K_T}(f_1 C_{\xi_1} - f_3 C_{\xi_3} + f_4 - f_2) \end{aligned} \tag{7}$$

where l is the distance from G to the rotor i and $K_M = 9 \times 10^{-6}$ N m s^2. The equality from Eq. (6) is ensured, meaning that:

$$\ddot{\eta} = \Pi_G(\eta)^{-1}[\tau - \dot{\Pi}_G(\eta)\dot{\eta}] \tag{8}$$

With $\tau = (\tau_\theta, \tau_\phi, \tau_\psi)^{\mathrm{T}}$ as an auxiliary inputs.

And

$$\Pi_G(\eta) = \begin{pmatrix} I_{xx} C_\phi & 0 & 0 \\ 0 & I_{yy} C_\phi C_\theta & 0 \\ 0 & 0 & I_{zz} \end{pmatrix} \tag{9}$$

As a first step, the model given above can be input/output linearized by the following decoupling feedback laws

$$\begin{aligned} \tau_\theta &= -I_{xx} S_\phi \dot{\phi}\dot{\theta} + I_{xx} C_\phi \tilde{\tau}_\theta \\ \tau_\phi &= -I_{yy} S_\phi C_\theta \dot{\phi}^2 - I_{yy} S_\theta C_\phi \dot{\theta}\dot{\phi} + I_{yy} C_\theta C_\phi \tilde{\tau}_\phi \\ \tau_\psi &= I_{zz} \tilde{\tau}_\psi \end{aligned} \tag{10}$$

and the decoupled dynamic model of rotation can be written as

$$\ddot{\eta} = \tilde{\tau} \tag{11}$$

with $\tilde{\tau} = (\tilde{\tau}_\theta \tilde{\tau}_\phi \tilde{\tau}_\psi)^{\mathrm{T}}$

Using the system of equations (2) and (11), the dynamic of the system is defined by

$$\begin{aligned} m\ddot{x} &= S_\psi C_\theta u_2 - S_\theta u_3 \\ m\ddot{y} &= (S_\theta S_\psi S_\phi + C_\psi C_\phi)u_2 + C_\theta S_\phi u_3 \\ m\ddot{z} &= (S_\theta S_\psi C_\phi - C_\psi S_\phi)u_2 + C_\theta C_\phi u_3 - mg \\ \ddot{\theta} &= \tilde{\tau}_\theta \\ \ddot{\phi} &= \tilde{\tau}_\phi \\ \ddot{\psi} &= \tilde{\tau}_\psi \end{aligned} \tag{12}$$

The rotational part can be easily linearized with static feedback control laws. Then, we get

$$\begin{aligned} \ddot{\theta} &= u_4 \\ \ddot{\phi} &= u_5 \\ \ddot{\psi} &= u_6 \end{aligned} \tag{13}$$

with

$$
\begin{aligned}
u_4 &= \frac{1}{I_{xx}C_\phi}(\tau_\theta + I_{xx}S_\phi\dot{\phi}\dot{\theta}) \\
u_5 &= \frac{1}{I_{yy}C_\theta C_\phi}(\tau_\phi + I_{yy}S_\phi C_\theta\dot{\phi}^2 + I_{yy}S_\theta C_\phi\dot{\theta}\dot{\phi}) \\
u_6 &= \frac{1}{I_{zz}}\tau_\psi
\end{aligned}
\tag{14}
$$

Remark. As shown in Eq. (2), the equivalent system of control-inputs presents five inputs $U = (u_2, u_3, u_4, u_5, u_6)$, while the rotor force-inputs are of six order $F = (f_1, f_2, f_3, f_4, \xi_1, \xi_3)$. Then the transformation $U \to F$ is not a diffeomorphism.

3. Advanced strategies of control

The aim in this section is to make comparison between model based approaches and experts analysis involving fuzzy systems. Classical model based techniques such us the *static feedback linearization* and *backstepping* techniques have been investigated and used for stabilization with motion planning [7,45,27].

3.1. Static feedback linearization controller

3.1.1. Control input for $z - y$motions

We propose to control the y/z motion through the input u_2/u_3. So, we have the following proposition.

Proposition. *Consider* $(\psi, \theta) \in]-\pi/2, \pi/2[$, *with the static feedback laws.*

$$
\begin{aligned}
u_2 &= m\nu_y C_\phi C_\psi^{-1} - m(\nu_z + g)S_\phi C_\psi^{-1} \\
u_3 &= m\nu_y(S_\phi C_\theta^{-1} - C_\phi tg_\psi tg_\theta) + m(\nu_z + g)(C_\phi C_\theta^{-1} + S_\phi tg_\psi tg_\theta)
\end{aligned}
\tag{15}
$$

The dynamic of y and z are linearly decoupled and exponentially asymptotically stable with the appropriate choice of the gain controller parameters. ν_y *and* ν_z *are detailed in the following.*

We can regroup the two dynamics as

$$
\begin{pmatrix} \ddot{y} \\ \ddot{z} \end{pmatrix} = \frac{1}{m}\begin{pmatrix} S_\theta S_\psi S_\phi + C_\psi C_\phi & C_\theta S_\phi \\ S_\theta S_\psi C_\phi - C_\psi S_\phi & C_\theta C_\phi \end{pmatrix}\begin{pmatrix} u_2 \\ u_3 \end{pmatrix} - \begin{pmatrix} 0 \\ g \end{pmatrix}
\tag{16}
$$

For the given conditions in ψ and θ the (2×2) matrix in Eq. (16) is invertible. Then a non-linear decoupling feedback permits to write the following decoupled linear dynamics:

$$
\begin{aligned}
\ddot{y} &= \nu_y \\
\ddot{z} &= \nu_z
\end{aligned}
\tag{17}
$$

Then we can deduce from Eq. (17) the linear controller:

$$
\begin{aligned}
\nu_y &= \ddot{y}_r - k_y^1(\dot{y} - \dot{y}_r) - k_y^2(y - y_r) \\
\nu_z &= \ddot{z}_r - k_z^1(\dot{z} - \dot{z}_r) - k_z^2(z - z_r)
\end{aligned}
\tag{18}
$$

with the k_y^i, k_z^i are the coefficients of a Hurwitz polynomial.

Our second interest is the (x, z) dynamics which can be also decoupled by a static feedback law.

3.1.2. Input u_2 for the motion along x

The aim here is to stabilize the dynamics of x with the control input u_2. While we keep the input u_3 for the altitude z stabilization.

Proposition. *With the new inputs*

$$
\begin{aligned}
u_2 &= (S_\psi C_\phi - S_\theta C_\psi S_\phi)^{-1}(m\nu_x C_\phi C_\theta + m(\nu_z + g)S_\theta) \\
u_3 &= (S_\psi C_\phi - S_\theta C_\psi S_\phi)^{-1}(-m\nu_x(S_\psi S_\theta C_\phi - C_\psi S_\phi) + m(\nu_z + g)S_\psi C_\theta)
\end{aligned}
\tag{19}
$$

The dynamic of x and z are decoupled and exponentially stable. ν_x and ν_z can be deduced as in [7]:

$$
\begin{aligned}
\ddot{x} &= \nu_x \\
\ddot{z} &= \nu_z
\end{aligned}
\tag{20}
$$

Remark. To switch between the two controllers Eqs. (15) and (19) continuity is recommended allowing the avoidance of the peak phenomenon. This can be asserted if we impose some constraints to the reference trajectories. In order to ensure this, we take $u_2(\psi = 0) = u_2(\psi = \pi/2) = 0$ with $\phi = \theta = 0$. For $\psi = 0$ one uses (Eq. (15)) and for $\psi = \pi/2$ expression Eq. (19) is considered. As soon as we get $u_3(\psi = 0) = u_3(\psi = \pi/2) = mg$ taking $\phi = \theta = 0$.

3.2. Self-tunable fuzzy inference system

The formal analogy between a fuzzy inference system and a multilayer neural network associated with optimization algorithms is used from the retro-propagation gradient algorithm have winded up in what is called a STFIS Network.

3.2.1. Presentation of STFIS

The main idea is to implement the various parts of a fuzzy logic system in a multilayered neural network. These operations can be traduced by the layer structure shown in Fig. 2. Each layer, connected with others by adjustable parameters, having a specific function [15,36]. Only output parameters of the FIS are optimized in this work.

The input layer receives the input signal. The first hidden layer calculates the membership grades μ_{aij} of the inputs to their associated fuzzy subsets so the weights a_{ij} allow to tune respectively the position and the width of the fuzzy subsets, the form of which is chosen through G_{ij}. The second one calculates the truth value α_i of each fuzzy rule using the *Min* operator. The output layer gives the crisp output value u by:

$$
u = \frac{\sum_{i=1}^{r}\alpha_i w_i}{\sum_{i=1}^{r}\alpha_i}
$$

where w_i are the synaptic weights between the second hidden layer and the output. So, basically, this kind of STFIS network implements a Sugeno's type fuzzy inference system where the weights w_i are the crisp conclusions of the rules. The optimization of the parameters of the network can be done by various methods. In this work, we use the backpropagation of the gradient to minimize the cost function.

3.2.2. Architecture and learning algorithm architecture

In this work, we propose to generate the fuzzy control rules by an optimization method, which is done entirely on-line. Jordan [24] proposes the *distal control* method, which is used by [34] under the name of JEAN (Jordan method Extended for Adaptive Neuro-control).

This architecture (Fig. 3, left) needs two STFIS networks:

(1) a first networks to identify the drone (Model);
(2) a second networks to control the drone (Controller).

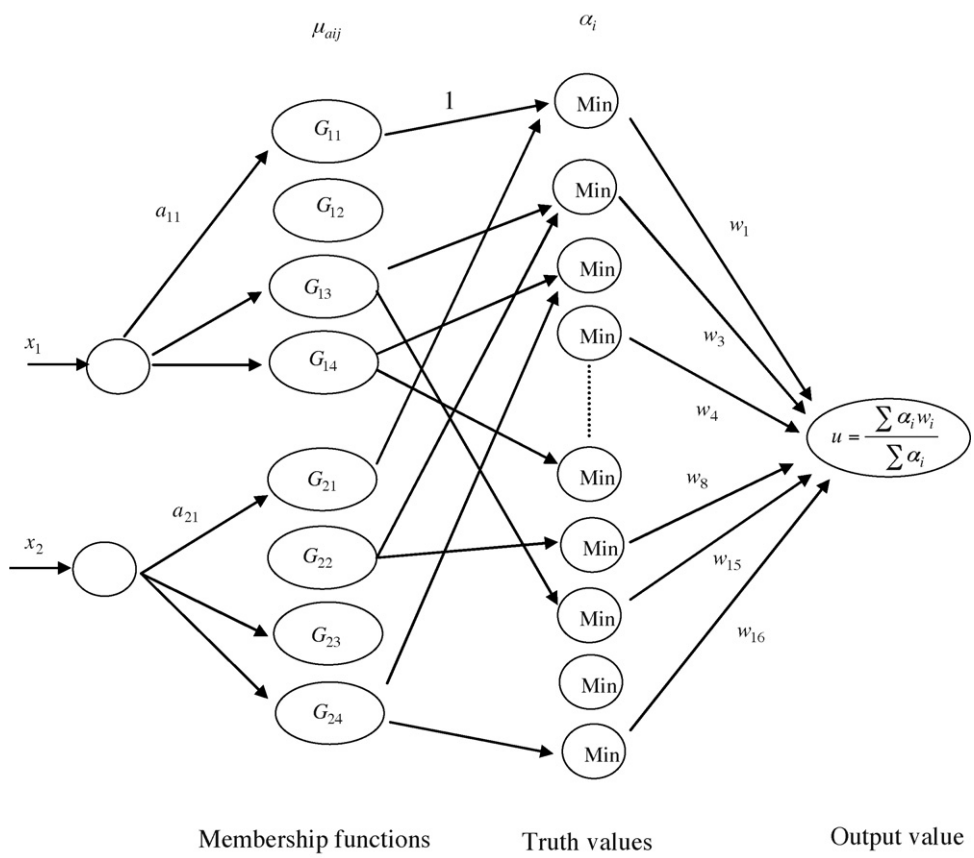

Fig. 2. Self-tunable fuzzy inference system.

The tuning of the synaptic weights is done by back-propagation of the quantity du^* through the network controller. The value of du^* itself is obtained by back-propagation of the cost function based on the output error e ($e = y - y_1$, where y_1 is the effective output value and y is the desired output) through the network model. But, to have a suitable control, it is not often necessary to know accurately the plant model (i.e. the *Jacobian*). In fact, an imprecise model of the plant does not alter the minimization of the cost function. Nevertheless, it modifies the itinerary taken by the learning algorithm in the action space: it will not follow the higher slope but it will stay on a downward path [24]. The technique using only the approximation of the *Jacobian* by its sign (easier to know then the *Jacobian* itself) is a viable approach to control simple (SISO) plants [35]. Then, we choose an architecture "mini-JEAN" as illustrated in Fig. 3(right) for the control of the XSF.

Compared to JEAN architecture, some similar performances are obtained in terms of generalization mean error. On the other hand, the computing time favors clearly mini-JEAN. Optimization of adjustable parameters is accomplished with a retro-propagation gradient algorithm which is adapted to the net structure (Fig. 3(right)). This architecture does not require an emulator network. It uses only one network as a controller, where the learning is done directly by the back-propagation of the output error. The simultaneity of the two phases can lead to serious problems of instability for JEAN architecture.

The aim is to minimize cost function E:

$$E = \tfrac{1}{2}\mathrm{e}^2 \tag{21}$$

where e is the difference between set point and process output. The basic equations of the algorithm are:

$$w_{ij}^n(t+1) = w_{ij}^n(t) + \Delta w_{ij}^n(t) \tag{22}$$

With

$$\Delta w_{ij}^n(t) = b\Delta w_{ij}^n(t-1) - \eta\delta_i^n\alpha_j^{n-1} - \frac{\beta\alpha_j^{n-1}w_{ij}^n(t-1)}{\sum_j \alpha_j^{n-1}} \tag{23}$$

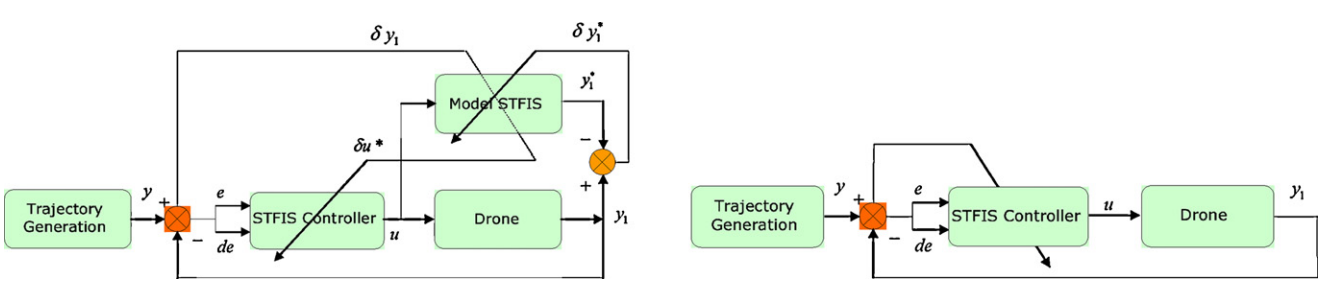

Fig. 3. JEAN learning architecture (left) and control architecture mini-JEAN (right).

where $w_{ij}^n(t)$ is the ith parameter between i of layer n and jth unit of layer $n-1$, η is the learning gain, t is the training iteration, b is the moment parameter, δ_i^n is the error term (ith neurone of layer n), α_j^{n-1} is the output of jth unit of layer $n-1$ and β is the regression coefficient. The quality of solution obtained using this algorithm depends on input learning signals, algorithm control parameters and learning duration (number of iterations).

3.2.3. Algorithm modification weight regression

The procedure is entirely done on-line on the engine. The table of rules (weights w_i) can be initially empty or filled with an a priori knowledge. The engine acquires by its systems output measures, calculates the error to the back-propagated, updates the triggered rules on-line. The weights of the table of decision are then adjusted locally and progressively.

The cost function is given by

$$J = E + \lambda \sum w_i^2 \tag{24}$$

where E is the classic quadratic error, w are the parameters (weights) to optimize parameters and λ is a constant that controls the growth of parameters. The second term in J is known as *weight decay* and used usually in the context of classification problems. This technique early used for autonomous wheeled robot [26] but, the authors have noticed, if the learning is prolonged, the weights increase continuously with time and progressively, the quality of the control deteriorates. To overcome this difficulty, we use here a well known technique in the field of neural classification methods [14] and having a strong relationship with ridge regression and regularization theory [9].

Thanks to the classic back-propagation algorithm, the parameters are modify as

$$w(t+1) = w(t) + b\Delta w(t) + \eta\left(\frac{-\partial J}{\partial w}\right) \tag{25}$$

This algorithm easily includes the effect of the second term of the cost function J and by taking $\beta = 2\lambda\eta$ (regression coefficient) we obtain:

$$w(t+1) = w(t) + b\Delta w(t) + \eta\left(\frac{-\partial E}{\partial w}\right) - \frac{\beta\alpha w(t)}{\sum \alpha} \tag{26}$$

Since a fuzzy inference system is concerned, we adapt this formula by multiplying β by the firing term of the rule, namely $\alpha_j / \sum \alpha_j$. α_j is the truth value of the premise part of the triggered rule.

By applying this method to our system, a saturation of weights growth is obtained without any degradation of the residual quadratic error and the quality of the control is maintained even under prolonged learning.

Table 1
Weight for z displacement

de \e	NB	NS	Z	PS	PB
NB	29.62	29.71	7.59	2.55	2.99
NS	29.52	31.34	10.63	3.02	1.69
Z	34.22	29.73	19.54	4.84	2.79
PS	39.72	41.37	22.32	1.74	9.61
PB	39.92	39.82	28.27	7.96	9.62

If we limit the optimization only on the conclusions parameters w_{1j}^4. Then, we get

$$\Delta w_{1j}^4(t) = b\Delta w_{1j}^4(t-1) - \eta\delta_1^4\alpha_j^3 - \frac{\beta\alpha_j^3 w_{1j}^4(t-1)}{\sum_j \alpha_j^3} \tag{27}$$

with

$$\delta_1^4 = \frac{e}{\sum_j \alpha_j^3} \tag{28}$$

4. Motion planning and simulation results

The XSF is tested in simulation in order to validate some motion planning algorithm considering the proposed STFIS control laws. We have considered a total mass equal to $m = 2$ kg. The technical characteristics of this flying vehicle were presented in [4]. We solve the tracking control problem using the point to point steering stabilization see [7,43] for more details.

A fuzzy controller based on an on-line optimization of a zero order Takagi–Sugeno fuzzy inference system is successfully applied. It is used to minimize a cost function that is made up of a quadratic error term and a weight decay term that prevents an excessive growth of parameters of the consequent part. The main idea is to generate the conclusion parts (so-called weight) of the rules automatically thanks to an optimization technique. The used method is based on a back-propagation algorithm where the parameters values are free to vary during the optimization process.

Starting with a preinitialized rules table, when XSF begins to fly, it performs the acquisition of the distances (observations), calculates the cost function to back-propagation, updates the triggered rules in real time, begins to move and so on. The weights w_i are then adjusted locally and progressively. The shape of the used membership functions is triangular and fixed in order to extract and represent the knowledge from the final results easily.

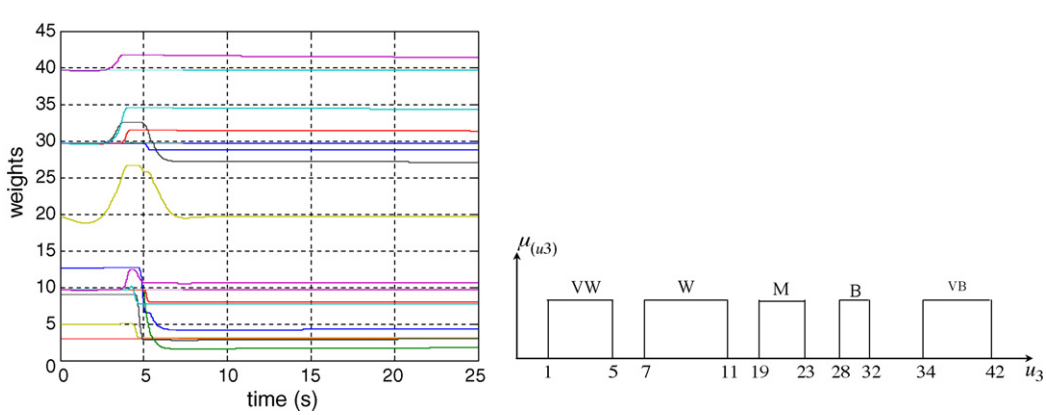

Fig. 4. Evolution of the weights for z displacement (left) and representation of the linguistic translation of the controller u_3 (right). (For interpretation of the references to colour in the artwork, the reader is referred to the web version of the article.)

Table 2
Learning linguistic table

de \e	NB	NS	Z	PS	PB
NB	B	B	W	VW	VW
NS	B	B	W	VW	VW
Z	VB	B	M	VW*	VW
PS	VB	VB	M*	VW*	W
PB	VB	VB	B	W	W

Table 3
Expertise linguistic table

de \e	NB	NS	Z	PS	PB
NB	B	B	W	VW	VW
NS	B	B	W	VW	VW
Z	VB	B	M	W	VW
PS	VB	VB	B	W	W
PB	VB	VB	B	W	W

The deduce the truth value, we use the MIN operator for the composition of the input variables. For the control of the XSF, we use the architecture known as "mini-JEAN". The universes of discourse are normalized and shared in five fuzzy subsets for all displacement.

The linguistic labels are defined as follows: NB: *Negative Big*, NS: *Negative Small*, Z: *approximately Zero*, PS: *Positive Small* and PB: *Positive Big*.

The results of the simulation are reported in the Table 1 for z displacement.

The optimization phase tends to stable weights (Fig. 4 left). In these circumstances the outputs linguistic labels could be interpreteted as follows (Fig. 4 right):

VW: [1, 5] *Very Weak*, W: [7, 11] *Weak*, M: [19, 23] *Medium*, B: [28, 32] *Big* and VB: [34, 42] *Very Big*.

Table 2 illustrates the linguistic translation of the table obtained by on-line optimization for the z displacement (Table 1).

By comparing the table proposed by learning and by human expertise (see Tables 2 and 3), we can observe that the two sets of linguistic rules are quite close. Three cases (noted with *) are different and they differ from only one linguistic concept (M instead B and VW instead W). So, we can claim that the extracted rules are quite logical and coherent. On the other hand, the main advantage of the described technique is the optimization of the controller with respect to the actual characteristics of the engine. The use of a function cost gathering a quadratic error and a term of regression of the weights enabled us to achieve our goal. For this behavior, the building of the navigation controller is done entirely on-line by the optimization of a zero order Takagi–Sugeno fuzzy inference system (FIS) by a back-propagation-like algorithm.

Figs. 5 and 6 (right) illustrates the controlled positions *zxy* using STFIS controller where u_3 and u_2, denote the command signals for z, x or y directions respectively. Note that the input $u_3 = mg$ at the equilibrium state is always verified. The inputs u_2 tend to zero after having carried out the desired orientation of the vehicle. It is also

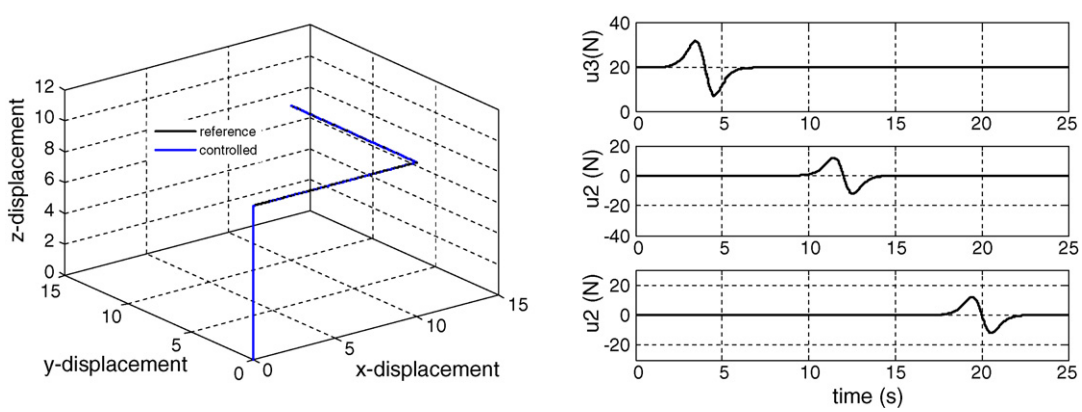

Fig. 5. Realization of a straight corners (left) and inputs u_3 and u_2 for the realization of a straight corners (right). (For interpretation of the references to colour in the artwork, the reader is referred to the web version of the article.)

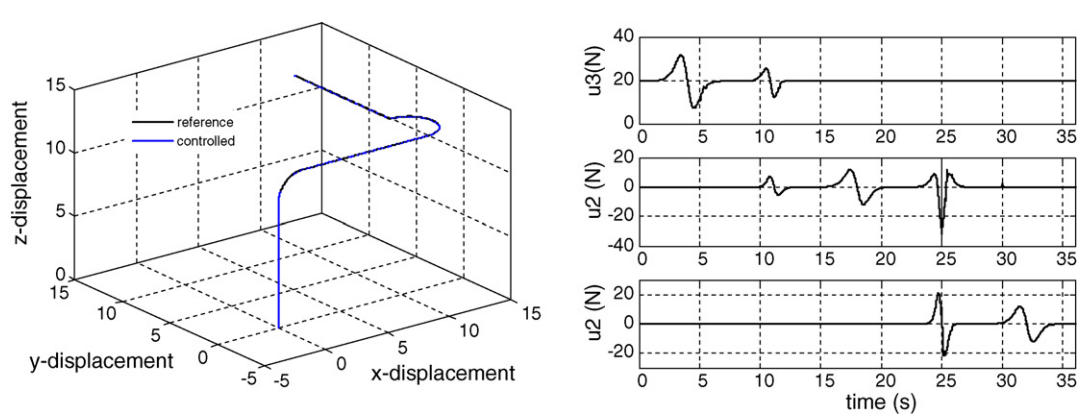

Fig. 6. Realization of a round corners (left) and inputs u_3 and u_2 for the realization of a round corners (right). (For interpretation of the references to colour in the artwork, the reader is referred to the web version of the article.)

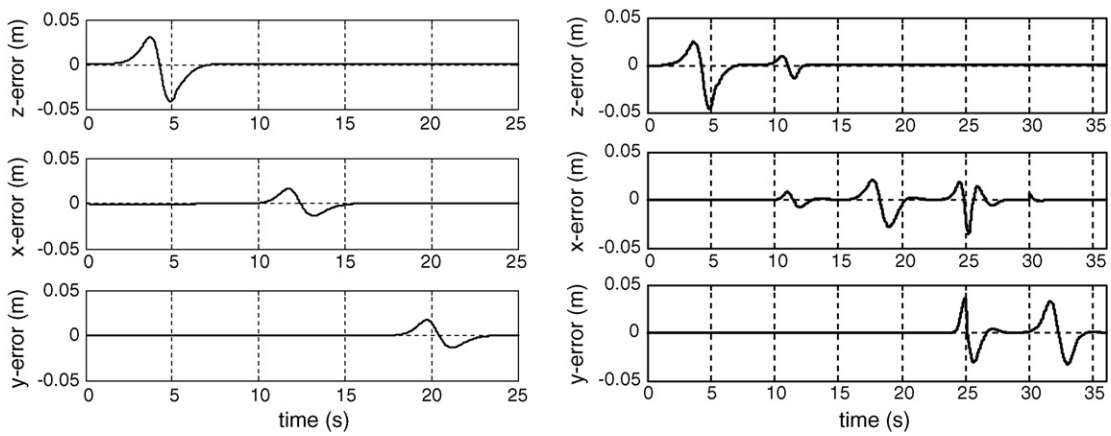

Fig. 7. Displacement errors of straight corners (left) and round corners (right).

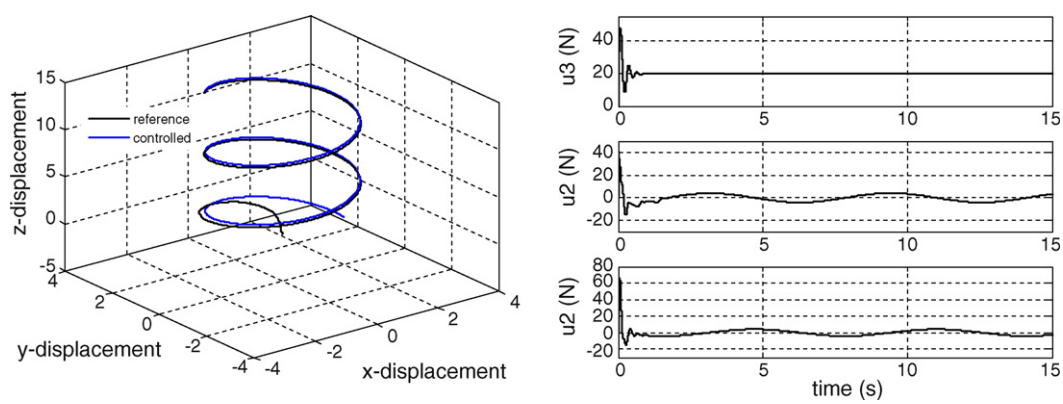

Fig. 8. Realization of a helical trajectory (left) and inputs u_3 and u_2 for the realization of a helical (right). (For interpretation of the references to colour in the artwork, the reader is referred to the web version of the article.)

shown in this figure that we can stabilize the system to make a following movement by the swivelling of the engine actuators 1 and 3.

The 3D displacement is depicted with straight, arc and round corners like connection (see Figs. 5 and 6(left)). These figures show the effectiveness of the used controller.

Fig. 7 shows displacement errors according to all the directions for the *X4 Stationnary Flyer*. It is noticed that the error thus tends to zero towards the desired positions.

According to Fig. 8 which represents the realization of a helical trajectory (left) and its input (right) that our controller ensures the trajectory continuation.

5. Controllers robustness

The robustness of the developed controllers are evaluated regarding external disturbances and performance degradations in the case of actuator/sensor failures and wind influence. In the case of the XSF, a resistance or a drag force is opposed to its movement in flight. The work produced by this force involves an additional energy consumption at the actuators levels which limits its maneuvering capacities in flight. This force can be expressed as follows:

$$F_i = \tfrac{1}{2} C_x \rho A V_i^2 \tag{29}$$

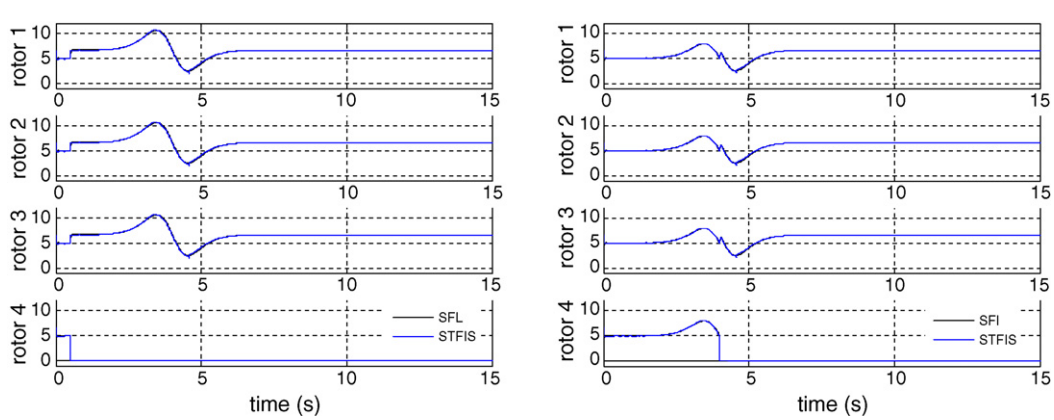

Fig. 9. XSF forces in the case of motor 4 failure at $t = 0.5$ s (left) and $t = 4$ s (right). (For interpretation of the references to colour in the artwork, the reader is referred to the web version of the article.)

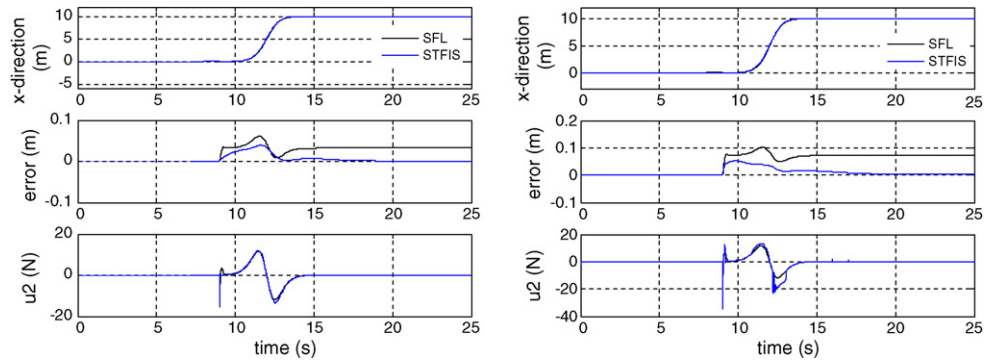

Fig. 10. Wind influence with a drag force of 1.4 N (left) and 2.1 N (right) for the x direction. (For interpretation of the references to colour in the artwork, the reader is referred to the web version of the article.)

where F_i [N] is the drag force following the i axis, V_i [m/s] is the drone velocity, A [m^2] is the coss-sectional area perpendicular to the force flow and ρ [kg/m^3] is the body density. The Eq. (29) induced a drag coefficient C_x which is a dimensionless quantity that describes a characteristic amount of aerodynamic drag depending on the XSF structure and which is experimently determined by windtunnel tests. This coefficient is equal to 0.5 for the x and y directions and 0.08 for the z displacement. The surface characteristic A of the XSF drone is equal to $A = 0.031\ \text{m}^2$ and it density is considered equal to $\rho = 1.22\ \text{kg/m}^3$.

Fig. 9 illustrate the simulation results in the case of actuator 4 failure after take-off at the instant $t_1 = 0.5$ s and $t_2 = 4$ s in the z direction. To maintain its equilibrium, the three remain actuators share the drone load compensation and which practically results in an equitable distribution of the developed forces ($F_1 + F_2 + F_3 = mg$ at steady state). The STFIS and the SFL controllers behaves in the same way.

Fig. 10 present the simulation results in the case of a drag force of $F_{\text{dg}} = 1.4$ N and of $F_{\text{dg}} = 2.1$ N according to the x displacement. The STFIS controller exhibits chattering signal problems in the transition phase while the SFL controller presents static errors that varies proportionally with the drag force amplitude F_{dg}. The same observations are found according to the two directions y and z.

6. Conclusion

In this paper, we studied a new configuration of flyer engine called XSF. We have considered in this work the stabilizing/tracking control problem for the three decoupled displacements of a XSF. The objectives are to test the capability of the engine to fly with straight, arc, rounded intersections and complex trajectories (helical).

We have presented and implemented an optimization technique allowing an on-line adjustment of the fuzzy controller parameters. The descent gradient algorithm, with its capacities to adapt to unknown situations by the means of its faculties of optimization, and the fuzzy logic, with its capacities of empirical knowledge modelling, are combined to control a new configuration of flyer engine. Indeed, we have obtained an on-line optimized Takagi–Sugeno type FIS of zero order. This method is simple, economical and safe since it is done on a mini-flying robots. It leads to very quick and efficient optimization technique. A comparison between the STFIS set rules and that deduced by human expertise, shows the validity of the proposed technique.

An analysis of the STFIS (which not require the good knowledge of the model) and the SFL (requires the well knowledge of the system model and parameters) controllers and their robustness regarding disturbances, shows the advantages and the disadvantages of these two techniques.

Future works will essentially investigate the real time implementation of the STFIS and the based-model control techniques. Obstacles avoidance and flying multi-drone are also envisaged thanks to the FIS faculties and its optimization capabilities.

References

[1] E. Altug, Vision based control of unmanned aerial vehicles with applications to an autonomous four rotor helicopter, Quadrotor, Faculties of the University of Pennsylvania, Pennsylvania, 2003.

[2] E. Altug, J.P. Ostrowski, R. Mahony, Control of a quadrotor helicopter using visual feedback, in: Proceeding of the IEEE International Conference on Robotics and Automation, Washington, DC, May, (2002), pp. 72–77.

[3] E. Altug, J.P. Ostrowski, C. Taylor, Quadrotor control using dual visual feedback, in: Proceeding of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan, September, (2003), pp. 4294–4299.

[4] N. Azouz, Y. Bestaoui, Modelling and simulation of a mini quad-rotor helicopter, in: Proceeding of DTM2005, ASME Design Theory and Methodology Coferences, Long Beach, CA, USA, September, 2005.

[5] K. Benzemrane, G.L. Santosuosso, G. Damm, Unmanned aerial vehicle speed estimation via nonlinear adaptive observers, in: American Control Conference, New York City, USA, July, (2007), pp. 11–13.

[6] L. Beji, A. Abichou, Streamlined rotors mini rotorcraft: trajectory generation and tracking, International Journal of Automation and Systems 3 (1 (March)) (2005) 87–99.

[7] L. Beji, A. Abichou, K.M. Zemalache, Smooth control of an X4 bidirectional rotors flying robots, in: Fifth International Workshop on Robot Motion and Control, Dymaczewo, Poland, June, 2005.

[8] Y. Bestaoui, R. Slim, Maneuvers for a Quad- Rotor Autonomous Helicopter, AIAA, Rohnert Park, California, 2007, pp. 7–10.

[9] C.M. Bishop, Regularization and complexity control in feedforward neural networks, in: Proceedings of the IEEE International Conference on Neural Networks, vol. 1, Paris, (1995), pp. 141–148.

[10] S. Bouabdallah, P. Murrieri, R. Siegwart, Design and control of an indoor micro quadrotor, in: IEEE International Conference on Robotics and Automation, New Orleans, USA, 2004.

[11] S. Bouabdallah, A. Noth, R. Siegwart, PID vs LQ control techniques applied to an indoor micro quadrotor, in: Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona, Spain, April, (2005), pp. 2259–2264.

[12] P. Castillo, A. Dzul, R. Lozano, Real-time stabilization and tracking of a four rotor mini-rotorcraft, IEEE Transactions on Control Systems Technology 12 (July (4)) (2004) 510–516.

[13] P. Castillo, R. Lozano, A. Dzul, Stabilization of a mini-rotorcraft having four rotors, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, (2004), pp. 2693–2698.

[14] M.Y. Chow, An analysis of weight decay as a methodology of reducing three-layer feedforward artificial neural network for classification problems, in: ICNN'94, Proceedings of the IEEE International Conference on Neural Networks, vol. 1, Orlando, (1995), pp. 600–605.

[15] P. Glorennec, An evolutive neuro-fuzzy network, in: Proceedings of the 4e International Conference "Neural Networks and their Applications", Nîmes, (1991), pp. 301–314.

[16] T. Hamel, R. Mahony, R. Lozano, J.P. Ostrowski, Dynamic modelling and configuration stabilization for an X4-flyer, in: IFAC 15th World Congress on Automatic Control, Barcelona, Spain, 2002.

[17] T. Hamel, R. Mahony, Pure 2D Visual Servo control for a class of under-actuated dynamic systems, in: Proceedings of the 2002 IEEE International Conference on Robotics and Automation, New Orleans, LA, (2004), pp. 2229–2235.
[18] J. Hauser, S. Sastry, G. Meyer, Nonlinear control design for slightly non-minimum phase systems: application to V/STOL aircraft, Automatica 28 (4) (1992) 665–679.
[19] S.D. Hanford, L.N. Long, J.F. Horn, A small semi-autonomous rotary-wing unmanned air vehicle (UAV), AIAA2005-7077.
[20] G. Hoffmann, D.G. Rajnarayan, S.L. Waslander, D. Dostal, J.C. Jang, C.J. Tomlin, The stanford testbed of autonomous rotorcraft for multi-agent control (STARMAC), in: Proceedings of the 23rd Digital Avionics System Conference, Salt Lake City, UT, November, 2004.
[21] G. Hoffmann, S.L. Waslander, Distributed cooperative search using information—theoretic costs for particle filters, with quadrotor applications, in: AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, CO, August, 2006.
[22] H. Ichihashi, Iterative fuzzy modelling and a hierarchical network, in: Proceedings of the 4th IFSA Congress, Brussels, vols. 49–52, (1991), pp. 21–24.
[23] J.S.R. Jang, ANFIS: adaptive-network-based fuzzy inference system, IEEE Transactions System, Man and Cybernet 23 (1993) 665–685.
[24] M.I. Jordan, D. Rumelhart, Internal world models and supervised learning, in: Proceedings of the eighth International Workshop on Machine Learning, 1991, pp. 70–74.
[25] H. Maaref, C. Barret, Progressive Optimization of a Fuzzy Inference System, IFSA-NAFIPS'2001, Vancouver, 2001, pp. 665–679.
[26] H. Maaref, C. Barret, Sensor-based navigation of a mobile robot in an indoor environment, Robotics and Autonomous Systems 38 (2002) 1–18.
[27] H. Maaref, K.M. Zemalache, L. Beji, Self-turnable fuzzy inference system: a comparative study for a drone, in: International Fuzzy Systems Association (IFSA), Cancun, Mexico, June 18–21, 2007.
[28] L. Mederreg, F. Diaz, N.K. M'sirdi, Nonlinear backstepping control with observer design for a 4 rotors helicopter, in: AVCS'04, International Conference on Advances in Vehicle Control and Safety, Genova, Italy, October, (2004), pp. 28–31.
[29] A. Mokhtari, A. Benallegue, Dynamic feedback controller of Euler angles and wind parameters estimation for a quadrotor unmanned aerial vehicle, in: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, 2004, pp. 2359–2366, April.
[30] A. Mokhtari, A. Benallegue, B. Daachi, Robust feedback linearization and GH_∞ controller for a quadrotor unmanned aerial vehicle, Journal of Electrical Engineering 57 (1) (2006) 20–27.
[31] V. Mistler, A. Benallegue, N.K. M'Sirdi, Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback, in: IEEE 10th IEEE International Workshop on Robot–Human Interactive Communication, Bordeaux and Paris, September, 2001.
[32] H. Nomura, I. Hayashi, N. Wakami, A self-tuning method of fuzzy control by descent method, in: Proceedings of the 4th IFSA Congress, Brussels, vols. 155–158, (1991), pp. 18–21.
[33] P. Pound, R. Mahony, P. Hynes, J. Roberts, Design of a four rotor aerial robot, in: Proceedings of the Australasian Conference on Robotics and Automation, Auckland, 2002, pp. 145–150.
[34] J.M. Renders, Mtaphore biologiques appliques á la commande de processus, Thse de doctorat, Universit libre de Bruxellese, Bruxellese, 1994.
[35] M. Saerens, Approche connexionniste de la commande de processus, Thse de Doctorat de l'Universit Libre de Bruxellese, Bruxellese, 1991.
[36] T. Takagi, M. Sugeno, Derivation of fuzzy control rules from human operator's control actions, in: Proceedings of the IFAC Symposium On Fuzzy Information, Knowledge Representation and Decision Analysis, 1983, pp. 55–60, July.
[37] A. Tayebi, S. McGilvray, Attitude stabilization of a VTOL quadrotor aircraft, IEEE Transactions on Control Systems Technology 14 (May (3)) (2006).
[38] G.P. Tournier, M. Valentiy ve, J.P. Howz, Estimation and control of a quadrotor vehicle using monocular vision and moiré patterns, in: AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, CO, USA, AIAA2006-6711, August, (2006), pp. 21–24.
[39] L.X. Wang, J.M. Rendel, Back-propagation fuzzy system as nonlinear dynamic system identifiers, in: Proceedings of the IEEE, International Conference on Fuzzy System, San Diego, (1991), pp. 1409–1416.
[40] S.L. Waslander, G. Hoffmann, J.S. Jang, C.J. Tomlin, Multi-agent X4-flyer testbed control design: integral sliding mode vs. reinforcement learning, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005, 468–473.
[41] C. Yang, L. Lai, C. Wu, Time-optimal control of a hovering quad-rotor helicopter, in: IEEE ICSS International Conference On Systems and Signals, 2005, 295–300.
[42] L.A. Young, E.W. Aiken, J.L. Johnson, R. Demblewski, J. Andrews, J. Klem, New concepts and perspectives on micro-rotorcraft and small autonomous rotary-wing vehicles, in: Proceedings of the 20th AIAA Applied Aerodynamics Conference, St. Louis, MO, USA, 2002.
[43] K.M. Zemalache, L. Beji, H. Maaref, Two inertial models of X4-flyers dynamics, motion planning and control, Journal of the Integrated Computer-Aided Engineering (ICAE) 14 (January 2) (2007) 107–119.
[44] K.M. Zemalache, L. Beji, H. Maaref, Backstepping control technique: application to an under-actuated X4-flyer, 151-156, in: Proceedings of the 1st International in Computer System and Information Technology Conference, vol. 1, Algeria, July, 2005.
[45] K.M. Zemalache, L. Beji, H. Maaref, Control of an under-actuated system: application to a four rotors rotorcraft, in: Proceedings of the 5th IEEE International Conference on Robotics and Biomimetics, vol. 1, Hong Kong, China, pp. 160–230, June 29–July 03, 2005.