Monocular Visual-Inertial SLAM-Based Collision Avoidance Strategy for Fail-Safe UAV Using Fuzzy Logic Controllers

Comparison of Two Cross-Entropy Optimization Approaches

**Changhong Fu · Miguel A. Olivares-Mendez ·
Ramon Suarez-Fernandez · Pascual Campoy**

Abstract In this paper, we developed a novel Cross-Entropy Optimization (CEO)-based Fuzzy Logic Controller (FLC) for Fail-Safe UAV to expand its collision avoidance capabilities in the GPS-denied environments using Monocular Visual-Inertial SLAM-based strategy. The function of this FLC aims to control the heading of Fail-Safe UAV to avoid the obstacle, e.g. wall, bridge, tree line et al, using its real-time and accurate localization information. In the Matlab Simulink-based training framework, the Scaling Factor (SF) is adjusted according to the collision avoidance task firstly, and then the Membership Function (MF) is tuned based on the optimized Scaling Factor to further improve the control performances. After obtained the optimal SF and MF, 64 % of rules has been reduced (from 125 rules to 45 rules), and a large number of real see-and-avoid tests with a quadcopter have done. The simulation and experiment results show that this new proposed FLC can precisely navigates the Fail-Safe UAV to avoid the obstacle, obtaining better performances compared to only SF optimization-based FLC. To our best knowledge, this is the first work to present the optimized FLC using Cross-Entropy method in both SF and MF optimization, and apply it in the UAV.

C. Fu (✉) · R. Suarez-Fernandez · P. Campoy
Computer Vision Group (CVG),
Centro de Automática y Robótica (CAR),
UPM-CSIC, Universidad Politécnica de Madrid,
C/ José Gutiérrez Abascal 2, 28006 Madrid, Spain
e-mail: fu.changhong@upm.es
URL: www.vision4uav.com

M. A. Olivares-Mendez
Automation Research Group,
Interdisciplinary Centre for Security,
Reliability and Trust, SnT-University of Luxembourg,
4 rue Alphonse Weicker, 2721, Luxembourg
URL: www.en.uni.lu/snt/

1 Introduction

The uncertainty, inaccuracy, approximation and incompleteness problems widely exist in real controlling techniques. However, the Fuzzy Logic Controller (FLC) as one of the most active and fruitful soft computing methods can well dealt with these issues. In addition, this *model-free* control approach often has the good robustness and adaptability in the highly nonlinear, dynamic, complex and time varying robot systems, e.g. Unmanned Aerial Vehicle (UAV).

The FLC is based on the fuzzy logic [45] that imitates human thinking and decision making with

natural language. Its essential part is a set of linguistic control rules, as shown in Tables 1, 2, 3, 4 and 5, related by the dual concepts of fuzzy implication and the compositional rule of inference. In other words, FLC provides an algorithm which can convert the linguistic control strategy based on expert knowledge into an automatic control method. It mainly consists of three different types of parameters: (I) Scaling Factor (SF), which is defined as the gains for inputs and outputs. Its adjustment causes *macroscopic* effects to the behavior of the FLC, i.e. affecting the whole rule tables; (II) Membership Function (MF), typically, it is the triangle-shaped function, as shown in Figs. 6–9, and its modification leads to *medium-size* changes, i.e. changing one row/column of the rule tables; (III) Rule Weight (RW), it is also known as the certainty grade of each rule, its regulation brings *microscopic* modifications for the FLC, i.e. modifying one unit of the rule tables. Zheng [47] presents a practical guide to tune FLC, which points out that the FLC can be manually tuned from *macroscopic* to *microscopic* effects, i.e. SF adjustment, MF modification and RW regulation.

The Cross-Entropy (CE) method derives its name from the Cross-Entropy (or Kullback–Leibler) distance, which is a fundamental concept of modern information theory. The method was motivated by an adaptive algorithm for estimating probabilities of rare events in complex stochastic networks, which involves variance minimization. In a nutshell, the CE method involves an iterative procedure where each iteration can be broken down into two phases. In the first stage, a random data sample (e.g. SF or MF of FLC) is generated according to a specified mechanism. Then, the parameters of the random mechanism are updated based on the data in order to produce a *better* sample in the next iteration. The CE method provides a unifying approach to simulation and optimization [40].

The localization techniques for UAV have obtained many promising performances, which use Global Positioning System (GPS), Motion Capture System (MCS), laser, camera, kinect (RGB-D Sensor) et al. However, considering the cost, size, power consumption, weight and surrounding information different sensors can

obtained steadily, camera is the best onboard option. It can achieve the Visual Odometry (VO) [42] to estimate the 6D pose of UAV. Monocular and stereo methods as the main camera-based approach are widely used in UAV, especially in quadrotor helicopters. But the stereo camera has its limitation when the baseline is much smaller than the distance between UAV and the target, and its cost, weight and power comsumption are higher (more heavy) compared to monocular camera. Thus, monocular camera is more popular and has been always researched recently, especially, many related works are presented using the advanced monocular Simultaneous Localization and Mapping (SLAM) [10] algorithms, they also overcame the drawback of monocular SLAM to estimate the real absolute scale to environments by fusing other sensors, e.g. Inertial Measurement Unit (IMU), in order to navigate the UAV accurately.

Collision avoidance (also referred to sense-and-avoid) problem has been identified as one of the most significant challenges facing the integration of aircraft into the airspace. Here, the term *sense* relates to the use of sensor information to automatically detect possible aircraft conflicts, whilst the term *avoid* relates to the automated control actions used to avoid any detected/predicted collisions [3]. The onboard single or multiple sensors can provide the sense-and-avoid capability for flying aircraft. However, as what has been mentioned above, the camera sensor is the best onboard candidate for UAV, which can be used in collision avoidance applications. Especially, Fail-Safe UAV requires this collision avoidance ability in the event of failures, e.g. GPS has dropped out, INS generated the drift, pilot sent the wrong control commands, the software/hardware of UAV has the faults suddenly et al.

Nonetheless, the main contribution of this paper are:

(I)   Developing the Robot Operating System (ROS)[1]-based FLC, which is a node providing three inputs and one output.

(II) Presenting the FLC training framework integrating with CE in Matlab Simulink,

---

[1]http://www.ros.org/

Table 1 Rules based on the *Zero* in the third input (integral of the error), *before* CE optimization

Dot error/error	Big Left	Left	Zero	Right	Big Right
Big Negative	Great Left	Big Left	Left	Little Left	Zero
Negative	Big Left	Left	Little Left	Zero	Little Right
Zero	Left	Little Left	Zero	Little Right	Right
Positive	Little Left	Zero	Little Right	Right	Big Right
Big Positive	Zero	Little Right	Right	Big Right	Great Right

Table 2 Rules based on the *Negative* in the third input (integral of the error), *before* CE optimization

Dot error/error	Big Left	Left	Zero	Right	Big Right
Big Negative	Big Left	Left	Little Left	Zero	Little Right
Negative	Left	Little Left	Zero	Little Right	Right
Zero	Little Left	Zero	Little Right	Right	Big Right
Positive	Zero	Little Right	Right	Big Right	Great Right
Big Positive	Little Right	Right	Big Right	Great Right	Great Right

Table 3 Rules based on the *Big Negative* in the third input (integral of the error), *before* CE optimization

Dot error/error	Big Left	Left	Zero	Right	Big Right
Big Negative	Left	Little Left	Zero	Little Right	Right
Negative	Little Left	Zero	Little Right	Right	Big Right
Zero	Zero	Little Right	Right	Big Right	Great Right
Positive	Little Right	Right	Big Right	Great Right	Great Right
Big Positive	Right	Big Right	Great Right	Great Right	Great Right

Table 4 Rules based on the *Positive* in the third input (integral of the error), *before* CE optimization

Dot error/error	Big Left	Left	Zero	Right	Big Right
Big Negative	Great Left	Great Left	Big Left	Left	Little Left
Negative	Great Left	Big Left	Left	Little Left	Zero
Zero	Big Left	Left	Little Left	Zero	Little Right
Positive	Left	Little Left	Zero	Little Right	Right
Big Positive	Little Left	Zero	Little Right	Right	Big Right

Table 5 Rules based on the *Big Positive* in the third input (integral of the error), *before* CE optimization

Dot error/error	Big Left	Left	Zero	Right	Big Right
Big Negative	Great Left	Great Left	Great Left	Big Left	Left
Negative	Great Left	Great Left	Big Left	Left	Little Left
Zero	Great Left	Big Left	Left	Little Left	Zero
Positive	Big Left	Left	Little Left	Zero	Little Right
Big Positive	Left	Little Left	Zero	Little Right	Right

which can be used as the *lazy* method to obtain the optimal SF and MF parameters of FLC.

(III) Applying this framework to solve a challenging task: collision avoidance for Fail-Safe UAV.

(IV) Designing a FLC-based Fail-Safe UAV with a monocular SLAM system.

(V) Optimizing two different kind of FLCs: (1) only SF is optimized in FLC (called SF-FLC) [33, 35]; (2) both SF and MF are optimized in FLC (named SFMF-FLC), and comparing their control performances in Fail-Safe UAV.

(VI) Summarizing the state-of-art works related to FLC optimization, UAV localization and its Collision Avoidance.

The outline of the paper is organized as follows: Section 2 introduced the related works. The monocular visual-inertial SLAM-based collision avoidance strategy is described in Section 3. Section 4 designed the FLC with its initial SFs, MFs and rule base. Then, the Cross-Entropy theory and its optimization method for FLC are introduced in Section 5. In Section 6, the UAV training framework and the optimized results are shown. In Section 7, the real flight results have been given and discussed. Finally, the concluding remarks and future work are presented in the Section 8.

2 Related Works

Fuzzy Logic Controller (FLC) as the promising controller for UAV has been researched in many literatures recently. The design process of a fuzzy logic based controller for a coaxial micro helicopter is presented in [22]. Doitsidis et al. [8] proposed a FLC for Unmanned Aerial Vehicle (UAV) to fly through specified waypoints in a 3D environment repeatedly, perform trajectory tracking, and duplicate/follow another vehicle's trajectory. Coza and Macnab [7] provides a new method to design the adaptive-fuzzy controller to achieve stabilization of a quadrotor helicopter in the presence of sinusoidal wind disturbance. Santos et al. [41] also implemented

a fuzzy controller to control the quadrotor. And [19] proved its FLC can obtain the expected performances in the UAV control and navigation tasks. However, all these FLCs are mainly designed and tested in the simulation softwares, e.g. Matlab, FlightGear Flight Simulator, GMS aircraft instrument.

Many FLCs have been developed and applied in the UAV real flights, [34] presents a FLC for UAV in a landing application. Olivares-Mendez et al. [32] proposes an implementation of two FLCs working parallelly for a pan-tilt camera platform on an UAV, but in these works, the parameters of FLC are manually adjusted in a huge amount of tests, this kind of regulation method for FLC not only requires the rich expert knowledge (exprience) and time, but also increases the risks in operating UAV.

Therefore, autonomous tuning/*lazy* methods for FLC are more competitive. A robustness comparison between model-based with self-tunable fuzzy inference system (STFIS) has been studied to control a drone in presence of disturbances in [18]. Kadmiry and Driankov [16] designed an gain scheduler-based FLC for an unmanned helicopter to achieve stable and robust aggressive maneuverability. An adaptive neuro-fuzzy inference system (ANFIS) based controller for UAV was developed to adjust its altitude, the heading and the speed together in [20]. The classical and multi-objective genetic algorithm (GA) based fuzzy-genetic autopilot are also designed and used for UAV in [4], which validated the time response characteristics, the robustness and the adaptation of fuzzy controller with respect to the large commands.

In [11], Haber et al. have proved that the Cross-Entropy (CE) is the best optimization technique for FLC, they use CE to tune the Scaling Factor (SF) of a PD fuzzy controller for cutting force regulation in a drilling process. And our previous works [33, 35] presented a CE-based optimization for SF in a PID fuzzy controller to command the UAV for avoiding a small obstacle with special color. Nonetheless, the CE in these works was limited to only optimizing the SF of FLC.

The UAV localization is a well studied problem in the robot community and has many practical applications. However, Global Positioning System

(GPS)-based flight [1] can not fly in the indoor environments and closely to the ground or urban canyons, where there is no/unstable GPS service. Motion Capture System (MCS)-based flying [39] just work in the indoor and local space where many expensive cameras with high speed are constructed. Laser Range Finder (LRF)-based flight [9] can not work in the open and broad places because of its limited detection distance. And Kinect (RGB-D Sensor)-based flying only works well in the room-sized indoor spaces [43]. Thus, with the improvement of *performance/price* in camera, many researchers are concentrating on vision-based localization, [6] presents the homography-based algorithms to compute the UAV relative translation and rotation by means of the images gathered by an onboard camera. Nemra and Aouf [29] proposed a robust airborne 3D Visual Simultaneous Localization and Mapping (VSLAM) solution based on a stereovision system. Bonak et al. [5] implemented a method to hold a quadcopter aircraft in a stable hovering position using artificial marker-based visual information and inertial measurement. But the monocular Simultaneous Localization and Mapping (SLAM) techniques are more popular recently, they has been researched in differnt UAV platforms, e.g. AscTec Products,[2] AR.Drone Parrot[3] et al., to incrementally build the consistent maps of their environments while simultaneously determining their location within these maps. Many UAV control and navigation works in Autonomous Systems Lab (ASL)[4] have presented different approaches to help monocular SLAM to estimate the real absolute scale to environments, e.g. Inertial Measurement Unit (IMU) as the proprioceptive measurement is used in monocular SLAM (exteroceptive information) by Weiss et al. [44]. Engel et al. [12] also proposed an method that enables a low-cost quadcopter, i.e. AR.Drone Parrot, to accurately fly various figures using visual and inertial information.

The civil and military applications for UAVs require the ability to plan collision-free trajec-tories to avoid the stationary obstacles, such as buildings, trees or unlevel ground. A survey of the collision avoidance approaches those deployed for unmanned aerial vehicles is presented in [2]. Merz and Kendoul [26] designed a LIDAR-based perception and guidance system that enables a helicopter to perform obstacle detection and avoidance, terrain following, and close-range inspection. A vision-based autonomous flight with a quadrotor type UAV is proposed and tested in the Google Earth virtual environment [13]. Lenz et al. [21] uses a single image captured from the onboard camera as input, produce obstacle classifications, and use them to select an evasive maneuver. Mejias and Campoy [23] presents a collision avoidance approach based on omnidirectional cameras that does not require the estimation of range between two platforms to resolve a collision encounter. Hrabar [15] presents a goal-directed 3D reactive obstacle avoidance algorithm for RUAVs to fly point-to-point type trajectories. He et al. [14] present a vision-based obstacle avoidance method using motion field information. Vision-based 3D geometry estimation for static obstacles has been proposed in [37] to control the UAV flying in the urban environments. Recchia et al. [38] have developed the optical flow-based approach to obtain the image depth to avoid the obstacles for UAV. Zufferey and Floreano [48] developed the autonomous micro-flyers capable of navigating within houses or small built environments. Especially, the Fail-Safe UAV systems, e.g. [30], often require the ablility of collision avoidance in the emergency situations with no crashes, or at least a minimum of damages caused by accidents.

3 Monocular Visual-Inertial SLAM-Based Collision Avoidance Strategy

3.1 Collison Avoidance Strategy

Many typical civil tasks, such as forest fire monitoring [25] in tree lines, fault inspection for buildings or bridges [28] in cluttered urban, are carrying out by UAV currently, and a field study after Hurricane Katrina [36] concluded one of the most important recommendations for autonomy
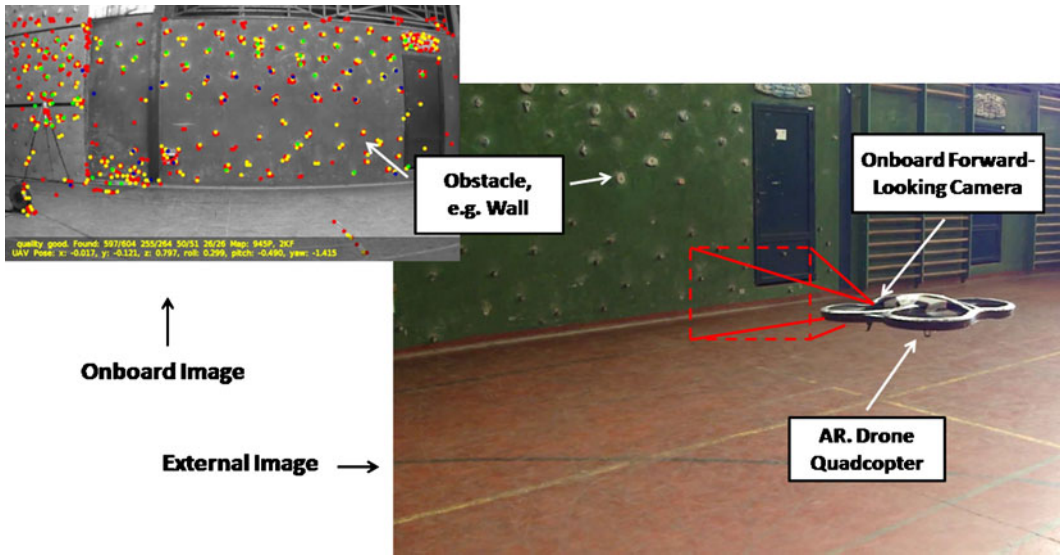

Fig. 1 Fail-Safe UAV is sensing its surrounding environment using monocular visual-inertial SLAM-based method. This real-time and accurate localization approach allows to expand the Fail-Safe UAV's collision avoidance capabilities in the event of failures. The details are described in Section 3

UAV is that the minimum emergent standoff distance from inspected structures is 2–5 m. This paper aims to discuss and research how to prevent crashes by UAV itself when UAV fly into this recommended distance based on the former working distance, i.e. Fail-Safe UAV avoids the collision in its (local) surrounding environment.

Considering a flying Fail-Safe UAV, e.g. AR. Drone Parrot, moving forwardly with a constant flight speed to an obstacle, e.g. wall, (Fig. 1) where, the heading of quadrotor helicopter is parallel to the normal vector of the obstacle. The con-

trol goal is to command it to avoid the obstacle, at least making it flying parallelly to the obstacle with a safe distance. Figure 2 shows the collision avoidance strategy, we divided the whole area into three parts:

(I)   Dangerous Alarm Area (DAA): it is set based on our quadcopter size (52.5 × 51.5 cm) and its inertance, as shown in the Fig. 3, this area is 1 meter in length;
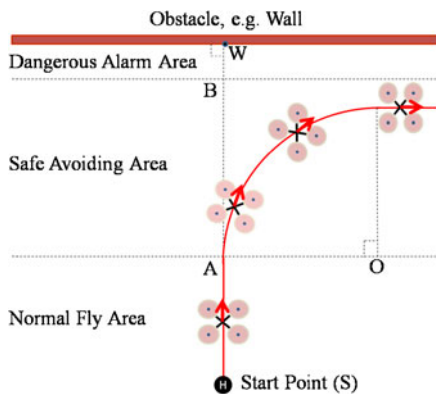
Fig. 2 2D description for collision avoidance task
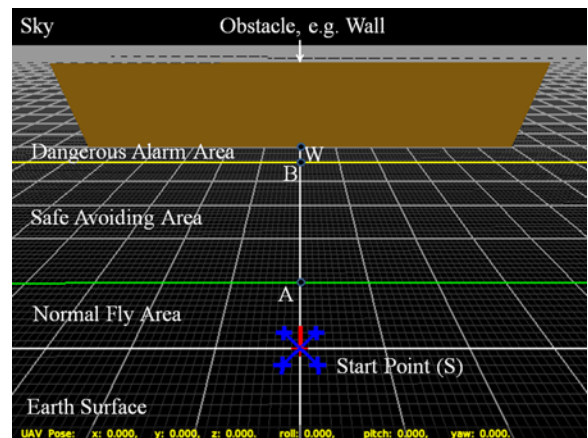
Fig. 3 Real-time 3D synchronization map, where, the scale of each big grid in white is equal to 1 meter in reality

(II) Safe Avoiding Area (SAA): it is designed for avoidance, which is based on the recommended distance from 1 meter to 4 meters in length;

(III) Normal Fly Area (NFA): it is the safe working area, the Start Point (S) can be set on any place in the NFA.

Therefore, for Fail-Safe UAV, once it flies into the SAA, the emergency control is enabled to prevent crash. And different constant flight speeds and sizes of SAA will be tested and evaluated in the simulations and real flights.

3.2 Monocular Visual-Inertial SLAM-Based Control

The real-time and accurate 6D pose estimations can provide the comprehensive and reliable information for Fail-Safe UAV during collision avoidance tasks, in the literatures, one single monocular camera and IMU sensor can be used to obtain this localization information. However, fusion of vision and IMU can be classified into 3 different categories [31]. The first section is named *Correction*, where it use the results from one kind of sensor to correct or verify the data from another sensor. The second category is *Colligation*, where one uses some variables resulting from the inertial data together with variables from the visual data. The third category is called *Fusion* and is by far the most popular method to efficiently combine inertial and visual data to improve pose estimation.
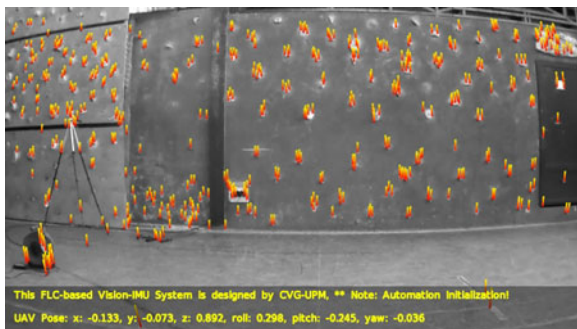
Fig. 4 FLC-based Fail-Safe UAV during automation initilization stage, where, the *orange-yellow line* stands for the tracked keypoint (FAST corners) movement from the first keyframe to current frame
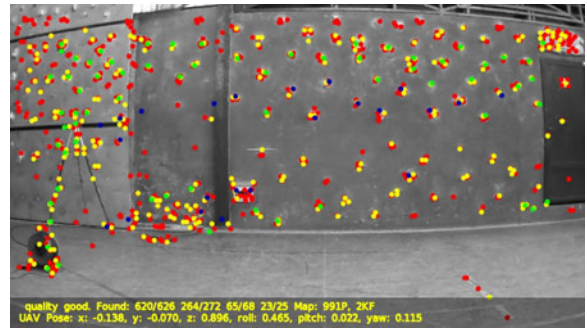
Fig. 5 FLC-based Fail-Safe UAV during visual fuzzy servoing stage, where, the *dot* represents the localization of keypoint. And the *colors* correspond to which pyramid-level the keypoint is measured in

Thus, in this paper, the *fusion* method is applied in monocular visual-inertial SLAM-based control for Fail-Safe UAV. For monocular vision, several visual odometry and visual SLAM frameworks have been launched in recent years. But the keyframe-based Parallel Tracking and Mapping (PTAM) [17] is more robust than filter-based SLAM, and it has the parallel processing threads and fast response performance. Besides, it is more suitable for common scenarios. It can provide 6 Degree of Freedom (DOF) estimation. For Inertial Measurement Unit (IMU), it is a 3D acceleration and rotation estimator.

As introduced in Section 2, Weiss et al. [44] presented that the Inertial Measurement Unit (IMU) as proprioceptive measurement can be used in monocular SLAM (exteroceptive information) to estimate the absolute scale, thereby navigating UAV precisely.[5] Engel et al. [12] has presented an Visual-Inertial SLAM system to enable UAV to precisely fly with different trajectories.[6] Both of them has published their works as the open sources in the Robot Operating System (ROS). Considering the UAV in our real flight tests, and the precise 6D pose estimation in the latter work, we developed a Fuzzy Logic Controller (FLC)-based Monocular Visual-Inertial SLAM system using his parts of codes, and all the parameters in this new system have been optimized. Figures 4 and 5 show the automation initialization of PTAM

---

[5]http://www.ros.org/wiki/asctec_mav_framework
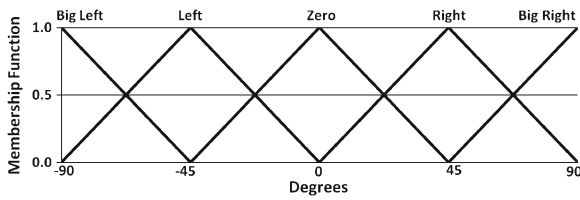[6]http://www.ros.org/wiki/tum_ardrone

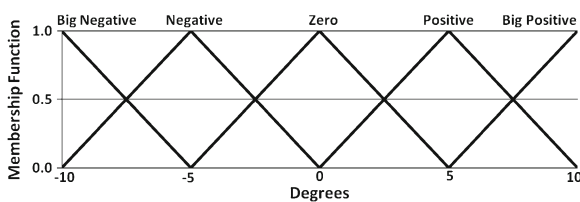**Fig. 6** Membership functions for the *first input* (yaw error), *without* CE optimization

Fig. 8 Membership functions for the *third input* (integral of yaw error), *without* CE optimization

and real-time image processing, which are fusing visual pose estimation with IMU measurement.

4 Fuzzy Logic Controller

Fuzzy Logic Controller (FLC) has the good robustness and adaptability in the highly nonlinear, dynamic, complex and time varying Fail-Safe UAV, thus, a FLC is designed to control its orientation. As previously developed FLCs in [33, 35], this FLC is developed in ROS based on the MOFS (Miguel Olivares' Fuzzy Software).

This FLC is a PID-like controller, which provides three inputs and one output. The first input is the angle error estimation in degrees between the angle reference (e.g. 90°) and the heading of Fail-Safe UAV. Other two inputs are the derivate and the integral values of this estimated angle error. The output is the command in degrees per seconds to change Fail-Safe UAV's heading. The initial Scaling Factors without CE optimization have the default value, which are equal to one. Since the collision avoidance task is identical for right or left side avoiding, and the Fail-Safe UAV has a symmetric design with the same behavior for left and right heading movements, this FLC has

the symmetric definitions in the inputs, output and the rule base.

Figures 6, 7, 8 and 9 show the initial definition for the Membership Functions in the inputs and output before Cross-Entropy (CE) Optimization. Each input has 5 sets, and the output has 9 sets. The symmetry of the FLC implies that any modification of the left side of each variable (input and output) can be applied to the right side directly.

The rule base was designed using the heuristic information based on expert knowledge. Each rule without CE optimization has a default weight, which is also equal to one. In other words, each rule has the same importance and effect to the FLC behavior. As shown in the definition of MFs, the three inputs in this FLC imply that the rule base has a cube construction, which is $5 \times 5 \times 5$ dimension. In this paper, 5 tables in 5×5 dimension are presented in order to show the rule base clearly. Each table is related to one of the 5 linguistic values of the third variable (i.e. the integral of error). Table 1 shows the rule base slide for the *Zero* value. Table 2 shows the slide for the *Negative* value. Table 3 shows the slide for the *Big Negative* value. Table 4 shows the slide for the *Positive* value. Finally, Table 5 shows the slide for the *Big Positive* value.
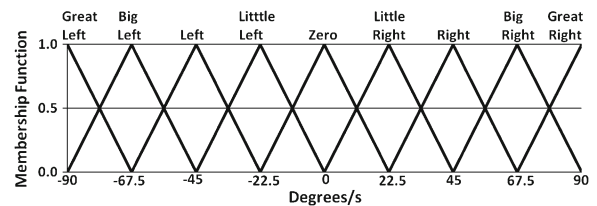

Fig. 7 Membership functions for the *second input* (derivative of yaw error), *without* CE optimization

Fig. 9 Membership functions for the *output* (yaw command), *without* CE optimization

The product t-norm is used for rules conjunction, and the defuzzification method used in this paper is a modification of the Height Weight method. Equation 1 shows the defuzzification method.

$$y = \frac{\sum_{l=1}^{M} \overline{y}^l \prod_{i=1}^{N} \left(\mu_{x_i^l}(x_i) \right)}{\sum_{l=1}^{M} \prod_{i=1}^{N} \left(\mu_{x_i^l}(x_i) \right)} \tag{1}$$

where, N and M represent the number of inputs variables and total number of rules, respectively. $\mu_{x_i^l}$ denotes the merbership function of the lth rule for the ith input variable. $\overline{y}^l$ represents the output of the lth rule.

5 Cross-Entropy Optimization

5.1 Optimization Principle

The CE method involves an iterative procedure where a random data sample $(x_1, \ldots, x_N)$ in the χ space is generated according to a specified random mechanism. A probability density function (pdf), such as the normal distribution, is used to update the data in order to produce a better sample in the next iteration. Let $g(-, v)$ be a family of probability density functions in χ, which is parameterized by a real value vector $v \in \Re$: $g(x, v)$. Let ϕ be a real function on χ, so the aim of the CE method is to find the minimum (as this paper proposed) or maximum of ϕ over χ, and the corresponding states x^* satisfying this minimum/maximum: $\gamma^* = \phi(x^*) = \min_{x \in \chi} \phi(x)$.

In each iteration, the CE method generates a sequence of $(x_1, \ldots, x_N)$ and $\gamma_1, \ldots, \gamma_N$ levels, such that γ converges to γ^* and x to x^*. Estimating the probability $l(\gamma)$ of an event $E_v = \{x \in \chi \mid \phi(x) \geq \gamma\}, \gamma \in \Re$ is concerned.

A collection of functions for $x \in \chi, \gamma \in \Re$ are defined:

$$I_v(x, \gamma) = I_{\{\chi(x_i) > \gamma\}} = \begin{cases} 1 & if \ \phi(x) \leq \gamma \\ 0 & if \ \phi(x) > \gamma \end{cases} \tag{2}$$

$$l(\gamma) = P_v(\chi(x) \geq \gamma) = E_v \cdot I_v(x, v) \tag{3}$$

where, E_v denotes the corresponding expectation operator.

In this manner, Eq. 3 transforms the optimization problem into an stochastic problem with very small probability. The variance minimization technique of importance sampling is used, in which the random sample is generated based on a pdf h. The sample $x_1, \ldots, x_N$ from an importance sampling density h on ϕ is evaluated by:

$$\hat{l} = \frac{1}{N} \cdot \sum_{i=1}^{N} I_{\{\chi(x_i) > \gamma\}} \cdot W(x_i) \tag{4}$$

where, $\hat{l}$ is the importance sampling and $W(x) = \frac{g(x, v)}{l}$ is the likelihood ratio. The search for the sampling density $h^*(x)$ is not an easy task, because the estimation of $h^*(x)$ requires that l should be known $h^*(x) = I_{\{\chi(x_i) > \gamma\}} \cdot \frac{g(x, v)}{l}$. So the referenced parameter v^* must be selected in the situation that the distance between h^* and $g(x, v)$ is minimal, therefore, the problem is simplified to a scalar case. The method used to measure the distance between these two densities is the Kullback–Leibler, also known as Cross-Entropy:

$$D(g, h) = \int g(x) \cdot ln \ g(x) dx - \int g(x) \cdot ln \ h(x) dx \tag{5}$$

The minimization of $D(g(x, v), h^*)$ is equivalent to maximize $\int h^* ln[g(x, v)] dx$, which implies $\max_v D(v) = \max_v E_p \left(I_{\{\chi(x) > \gamma\}} \cdot ln \ g(x, v) \right)$, in terms of importance sampling, it can be rewritten as:

$$\max_v \hat{D}(v) = \max \frac{1}{N} \sum_{i=1}^{N} I_{\{\chi(x_i) > \gamma\}}$$
$$\cdot \frac{p_x(x)}{h(x_i)} \cdot ln \ g(x_i, v) \tag{6}$$

where, h is still unknown, therefore, the CE algorithm will try to overcome this problem by constructing an adaptive sequence of the parameters $(\gamma_t \mid t \geq 1)$ and $(v_t \mid t \geq 1)$.

5.2 FLC Optimization Description

The Cross-Entropy method generates N FLCs: $x_i = (x_{i1}, x_{i2}, \ldots, x_{ih})$, where, $i = 1, 2, .., N$, h represents the number of optimization objects. The

probability density functions: $g(x, v) = (g(x_1, v), g(x_2, v), \ldots, g(x_h, v))$ are also generated from CE method. Then, the objective function values are calculated for each FLC. The optimization objects: $x_{i1}, x_{i2}, \ldots, x_{ih}$ in the first optimization stage correspond to SFs, i.e. K_p, K_d and K_i, and then they represent the position and size of the membership function sets in the second stage. After all the generated controllers are tested in one iteration, the $g(x, v)$ is updated using a set of best FLCs (i.e. elite FLCs). The number of elite FLCs used to update the pdf is denoted as N^{elite}. Then, new N FLCs are generated, which are tested in the next iteration. When the maximum number of iteration is reached, the optimization process is over. A generic version of the optimization process for FLCs is presented in the Algorithm 1.

Algorithm 1 Cross-Entropy Algorithm for Fuzzy controller optimization

---

1. Initialize $t = 0$ and $v(t) = v(0)$
2. Generate N FLCs: $(x_i(t))_{1 \leq i \leq N}$ from $g(x, v(t))$, being each $x_i = (x_{i1}, x_{i2}, \ldots, x_ih)$
3. Compute $\phi(x_i(t))$ and order $\phi_1, \phi_2, \ldots, \phi_N$ from smallest ($j = 1$) to biggest ($j = N$). Get the N^{elite} first controllers $\gamma(t) = \chi_{[N^{\text{elite}}]}$.
4. Update $v(t)$ with $v(t+1) = arg_v \min \frac{1}{N^{\text{elite}}} \sum_{j=1}^{N^{\text{elite}}} I_{\{\chi(x_i(t)) \geq \gamma(t)\}} \cdot \ln g(x_j(t), v(t))$
5. Repeat from step 2 until convergence or ending criterion.
6. Assume that convergence is reached at $t = t^*$, an optimal value for ϕ can be obtained from $g(., v(t)^*)$.

In these two optimization processes, the gaussian distribution function was used. The mean μ and the variance σ of h parameters are calculated for each iteration as $\tilde{\mu}_{th} = \sum_{j=1}^{N^{\text{elite}}} \frac{x_{jh}}{N^{\text{elite}}}$, $\tilde{\sigma}_{th} = \sum_{j=1}^{N^{\text{elite}}} \frac{(x_{jh} - \mu_{jh})^2}{N^{\text{elite}}}$. The mean vector $\bar{\tilde{\mu}}$ should converge to γ^* and the standard deviation $\bar{\tilde{\sigma}}$ to zero, where, $N^{\text{elite}} = 5$.

In order to obtain the smooth updates for the mean and variance in iterations, a set of parameters, i.e. α, η, β in Eq. 7, have been applied, where α is a constant value used for the mean, η is a variable value, which is applied to the variance to avert the occurrences of $0s$ and $1s$ in the parameter

vectors, and β is a constant value, which changes the value of $\eta(t)$.

$$\eta(t) = \beta - \beta \cdot \left(1 - \frac{1}{t}\right)^q$$

$$\hat{\mu}(t) = \alpha \cdot \tilde{\mu}(t) + (1 - \alpha) \cdot \hat{\mu}(t-1)$$

$$\hat{\sigma}(t) = \eta(t) \cdot \tilde{\sigma}(t) + (1 - \eta(t)) \cdot \hat{\sigma}(t-1) \quad (7)$$

where, $\hat{\mu}(t-1)$ and $\hat{\sigma}(t-1)$ are the previous values of $\hat{\mu}(t)$ and $\hat{\sigma}(t)$. The values of the smooth update parameters are set: $0.4 \leq \alpha \leq 0.9$, $0.6 \leq \beta \leq 0.9$ and $2 \leq q \leq 7$. In order to get an optimized controller, the objective function named Integral Time of Square Error (ITSE) is selected.

6 Training Framework and Optimized Results

6.1 Training Framework

The training framework developed in Matlab Simulink is presented in this section, which is used to optimize the FLC to change Fail-UAV's heading. In this framework, the optimization will be processed by itself, and each controller generated by the Cross-Entropy method is tested. The main simulink blocks include UAV model, virtual camera, obstacle and FLC.

6.1.1 UAV Model

This *UAV Model* block is designed for AR. Drone Parrot, Asctec Pelican and LinkQuad[7] in CVG-UPM[8] according to [24] and [27], as shown in the Fig. 10. This block has four different types of input commands, however, the presented avoiding task only require to control the orientation of Fail-Safe UAV, and it will move forwardly with a constant flight speed to an obstacle, thus, the pitch and yaw commands are only to be controlled, i.e. roll and altitude commands were set to 0. For the UAV pitch commands, different constant speeds were sent in all the tests in order to improve the generalization performance of FLC. For the yaw

---

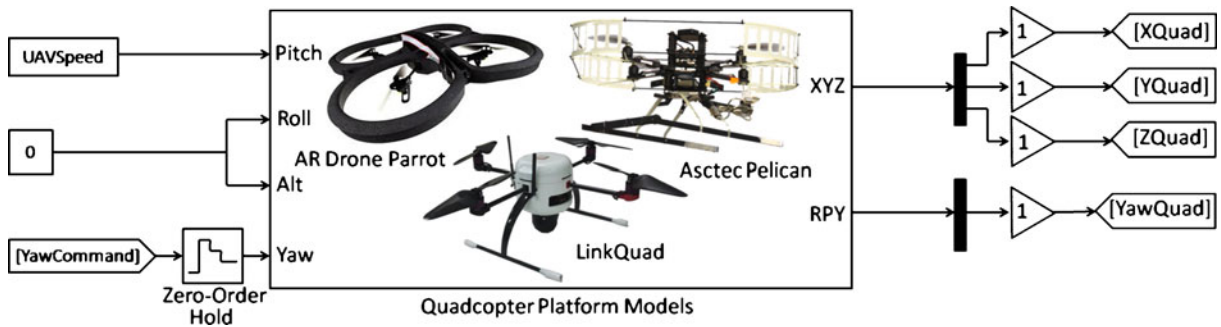[7] http://www.uastech.com/

[8] http://www.vision4uav.com

Fig. 10 UAV model block in Matlab Simulink, where, it is suitable for different quadcopter platforms

commands, they are generated by the optimizing FLCs. The outputs of this block are the current 6D pose of the quadcopter.

6.1.2 Obstacle

The *Obstacle* block is implemented for Fail-Safe UAV to avoid, as shown in the Fig. 11. This block contains six inputs: the initial positions, i.e. x, y, z of obstacle, linear speed, angular speed and the orientation. In the collision avoidance task, the obstacle, e.g. wall, is static, thus, the speed commands are set to 0. In the other hand, the heading of UAV is parallel to the normal vector of the obstacle at the beginning of task, therefore, the orientation is also given to 0. The x position of obstacle is set according to the minimum emergent standoff distance mentioned in [36], in our work, it is equal to 5. Additionally, the control goal is to command the Fail-Safe UAV to avoid the obstacle and fly parallelly to the obstacle, thus, the boundary of obstacle is set to a big number, e.g. 10^6.

6.1.3 Virtual Camera

The *virtual camera* block is constructed to simulate the onboard forward camera in Fail-Safe UAV, which is used to detect the target (i.e. obstacle) and provide the angle reference information to the FLC. The Fig. 12 shows the onboard virtual camera block implemented in Matlab Simulink. The inputs of this block includes the current positions of the target and the quadcopter. The output is the horizontal angle, which is equal to 90° according to the above parameters set for obstacle.

6.1.4 Fuzzy Logic Controller

Figure 13 is the Fuzzy Logic Controller (FLC) block for controlling the heading of Fail-Safe UAV. It has 3 inputs and 1 output. The three inputs are the yaw error, its derivate and integral values. And the initial membership functions and rule base are set to it based on the Tables 1–5 and Figs. 6–9. And the initial scaling factors are set
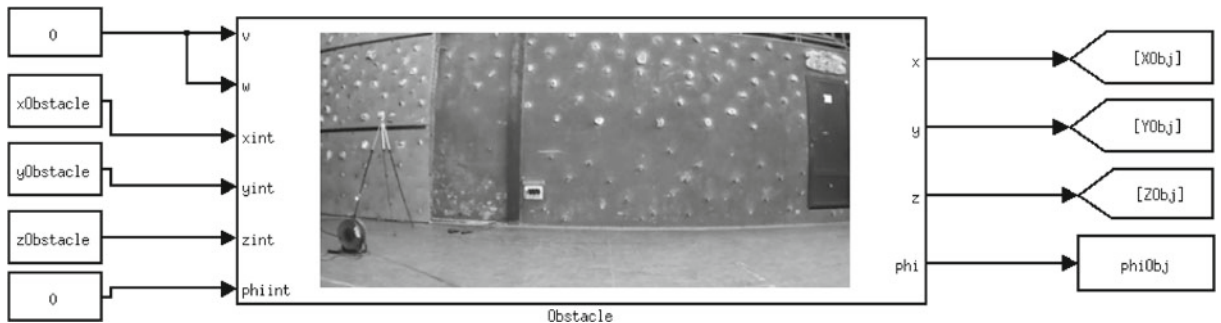
Fig. 11 Obstacle block in Matlab Simulink, where, the wall as the obstacle is tested
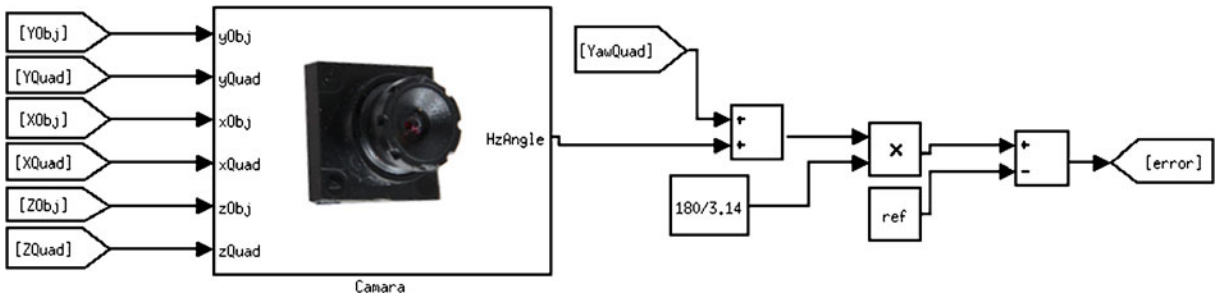
Fig. 12 Virtual camera block in Matlab Simulink, where, the AR. Drone Parrot is used as the Fail-Safe UAV
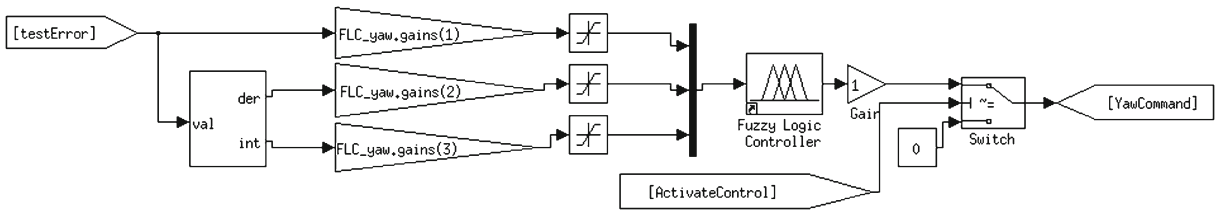

Fig. 13 Fuzzy Logic Controller (FLC) block in Matlab Simulink. It is applied to control Fail-Safe UAV's heading
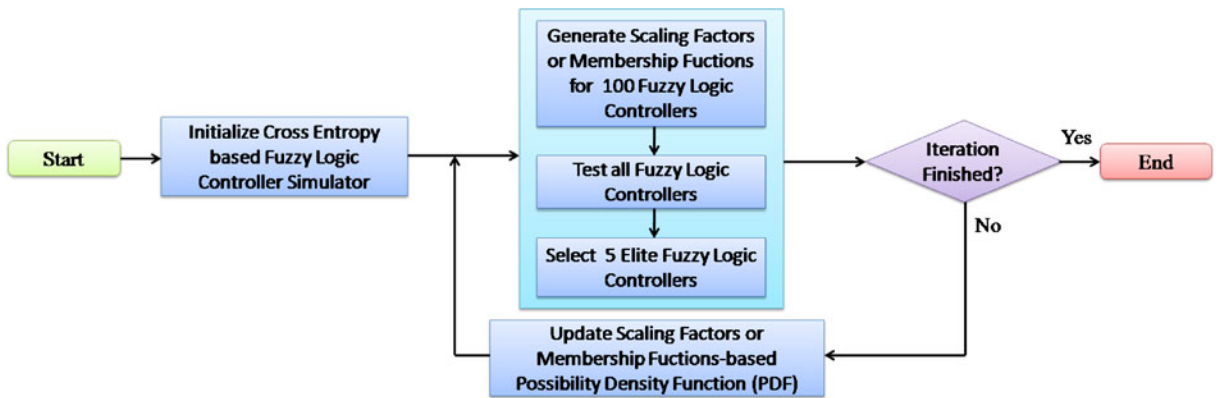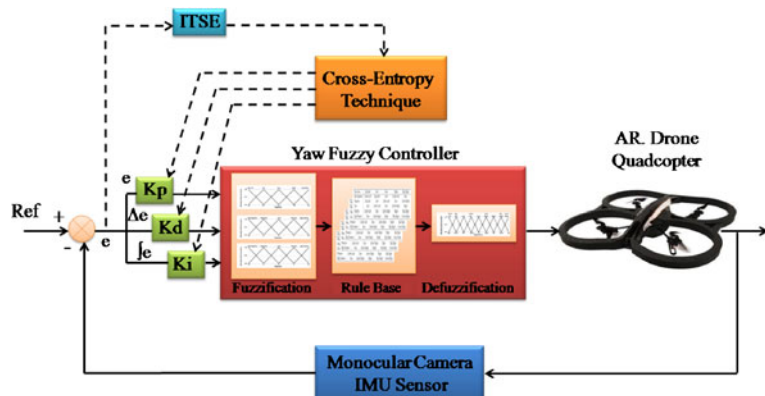

Fig. 14 Flowchart of Cross-Entropy optimization for Scaling Factors and Membership Functions in FLCs

Fig. 15 Cross-Entropy optimization for Scaling Factors in FLC

to 1. This FLC generates the yaw command to the Fail-Safe UAV to avoid the collision.

6.2 Optimized Results

In the literatures, although CE has been used to optimized many FLCs in different systems, their optimized parameters are only limited to Scaling Factors. In this paper, an new application of this method was presented for optimizing both Scaling Factor (SF) and Membership Function (MF) in the FLC as mentioned above, i.e. a Macroscopic and Medium-size optimization are presented.

The initial parameters for Cross-Entropy method were set based on [11, 33] and [46]. Figure 14 shows the whole training process.

6.2.1 Scaling Factors Optimization Results

Figure 15 shows the control loop during the scaling factors, i.e. K_p, K_d, K_i, optimization stage. The evolution can be shown with mean and sigma values associated with each scaling factor. Both values can be used to represent the Probability Density Function (PDF) in each iteration.

Figure 16 shows the evolution of the PDF for the SF of first input in FLC, its optimized value is 4.6739. Similarly, the optimal scaling factors for second and third input, as shown in Figs. 17

Fig. 17 The evolution of the PDF for the Scaling Factor of second input (K_d) in FLC using CE method. The optimal Scaling Factor for second input is 0.03

and 18, are 0.03 and −0.5003, respectively. In the SFs optimization stage, its winner iteration is the 85th iteration in 100 iterations.

6.2.2 Membership Functions Optimization Results

After obtained the optimal SFs (K_{po}, K_{do}, K_{io}) for FLC, the Membership Functions should be

Fig. 16 The evolution of the PDF for the Scaling Factor of first input (K_p) in FLC using CE method. The optimal Scaling Factor for first input is 4.6739
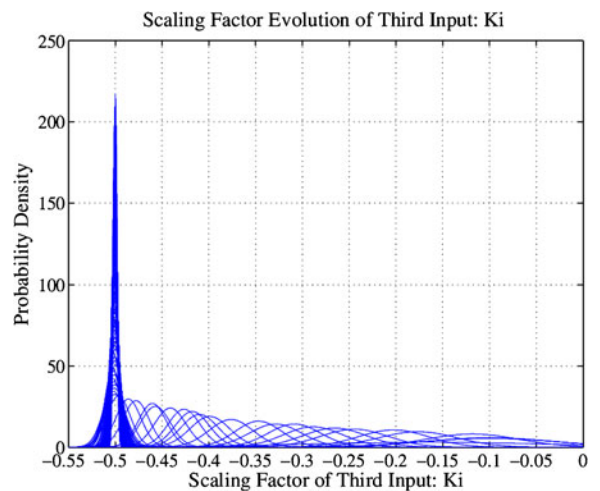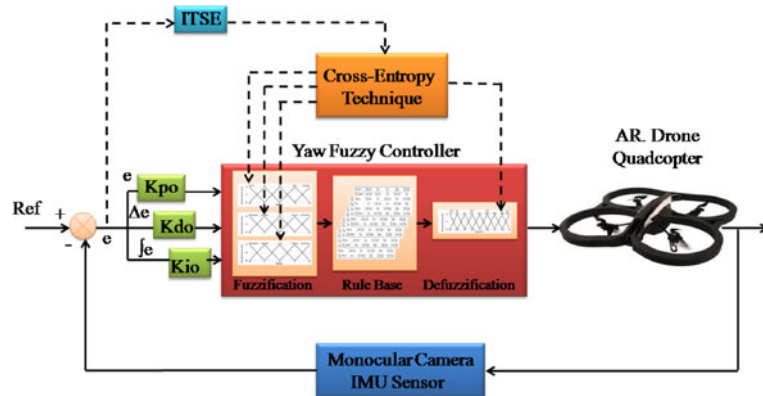

Fig. 18 The evolution of the PDF for the Scaling Factor of third input (K_i) in FLC using CE method. The optimal Scaling Factor for third input is −0.5003

Fig. 19 Cross-Entropy
optimization for
Membership Functions
based on the optimized
Scaling Factors in FLC

optimized. Figure 19 shows the control loop during the membership functions optimization stage. Considering the membership functions are symmetric, any position modification of the left side of each variable (input and output) can be applied to the right side directly.

Figure 20 shows the evolution of the PDF for the membership functions of first input (*Left*) of FLC, its optimized value is −89.6960, thus, the optimal membership function for first input (*Right*) is 89.6960. Similarly, the optimal membership function for second input (*Negative*), third input (*Negative*) are −8.1166 and −9.9782, respectively. Thus, the optimal membership function for

second input (*Positive*), third input (*Positive*) are 8.1166 and 9.9782, respectively. The evolution of the PDF for second and third input have been shown in Figs. 21 and 22.

Figure 23 shows the evolution of the PDF for the membership functions of output (*Big Left*) of FLC, its optimized value is −88.974, thus, the optimal membership function for output (*Big Right*) is 88.974. Similarily, the optimal membership function for output *Left* and *Little Left* are −88.191 and −74.952, respectively. Hence, the optimal membership function for output *Right* and *Little Right* are 88.191 and 74.952, respectively. The evolution of the PDF for output *Left* and *Little*
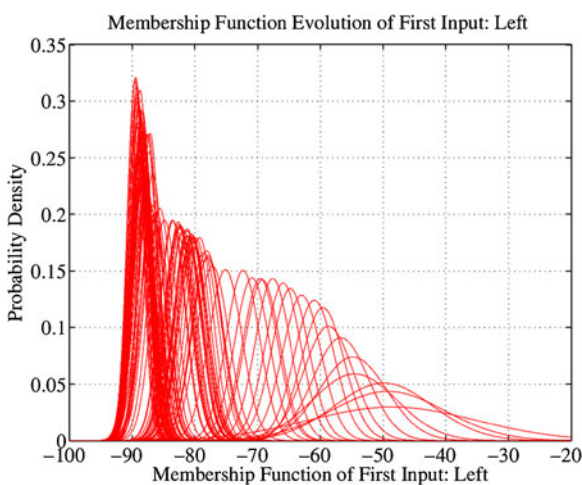

Fig. 20 The evolution of the PDF for the Membership Function of first input (*Left*) in FLC using CE method. The optimal Membership Function for *Left* is −89.6960, then, the optimal *Right* is 89.6960
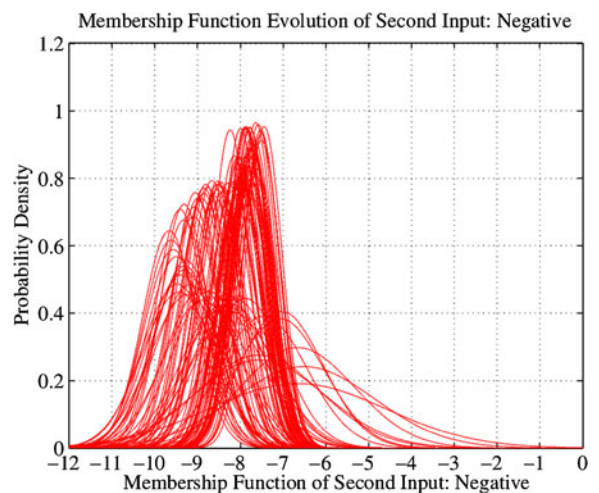

Fig. 21 The evolution of the PDF for the Membership Function of second input (*Negative*) in FLC using CE method. The optimal Membership Function for *Negative* is −8.1166, then, the optimal *Positive* is 8.1166
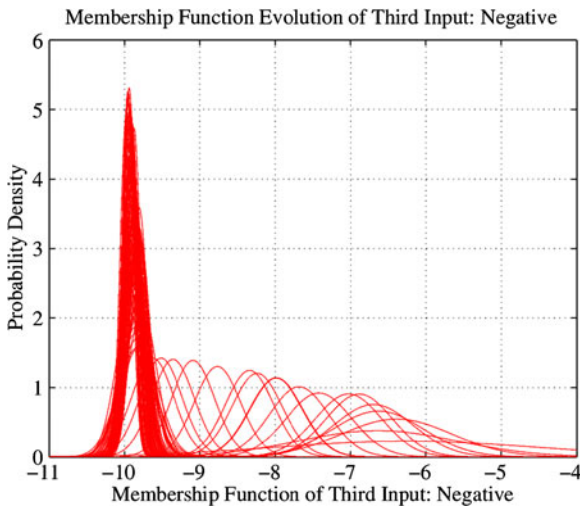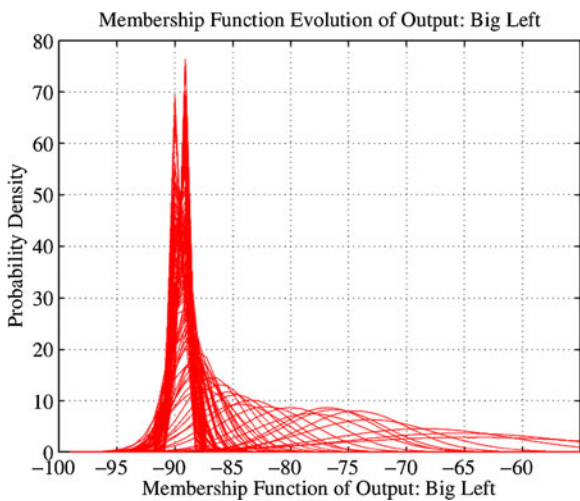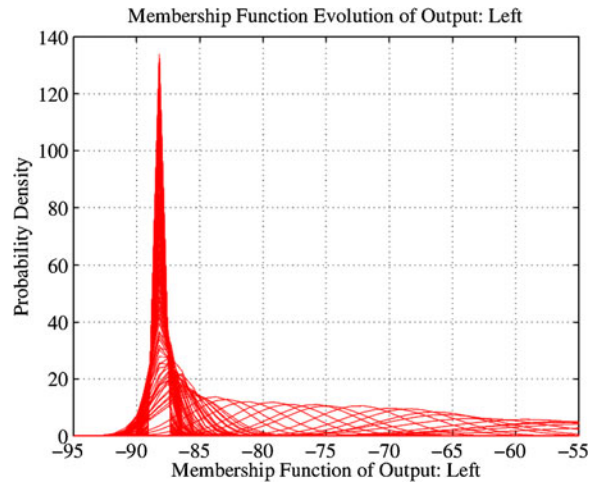
Fig. 22 The evolution of the PDF for the Membership Function of third input (*Negative*) in FLC using CE method. The optimal Membership Function for *Negative* is −9.9782, then, the optimal *Positive* is 9.9782

Left also have been shown in Figs. 24 and 25. In the MFs optimization stage, its winner iteration is the 93th iteration in 100 iterations.

After CE-based optimization for MFs, the new MFs have been generated for FLC, as shown in the Figs. 26, 27, 28 and 29, where, some sets of membership functions are nearly overlapped

Fig. 23 The evolution of the PDF for the Membership Function of output (*Big Left*) in FLC using CE method. The optimal Membership Function for *Big Left* is −88.974, then, the optimal *Big Right* is 88.974

Fig. 24 The evolution of the PDF for the Membership Function of output (*Left*) in FLC using CE method. The optimal Membership Function for *Left* is −88.191, then, the optimal *Right* is 88.191.
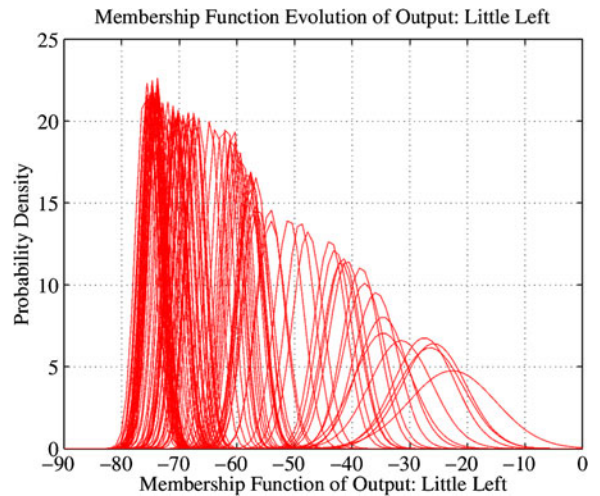

Fig. 25 The evolution of the PDF for the Membership Function of output (*Little Left*) in FLC using CE method. The optimal Membership Function for *Little Left* is −74.952, then, the optimal *Little Right* is 74.952
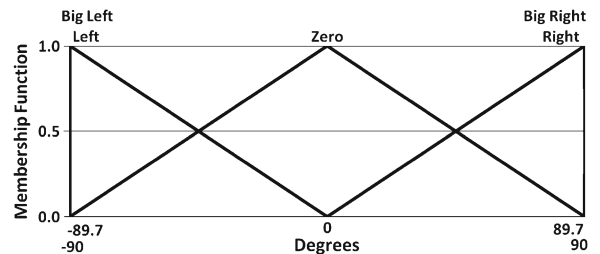

Fig. 26 Membership functions for the *First input* (Yaw Error), *after* CE optimization, where, the *Left* (*Right*) has been optimized to −89.6960 (89.6960) compared to the Fig. 6
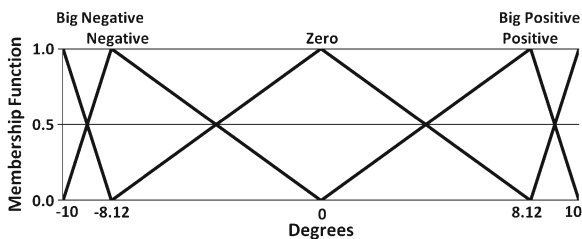
Fig. 27 Membership functions for *Second input* (Derivative of Yaw Error), *after* CE optimization, where, the *Negative* (*Positive*) has been optimized to −8.1166 (8.1166)
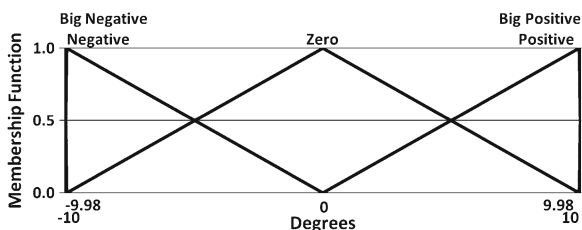

Fig. 28 Membership functions for *Third input* (Integral of Yaw Error), *after* CE optimization, where, the *Negative* (*Positive*) has been optimized to −9.9782 (9.9782)
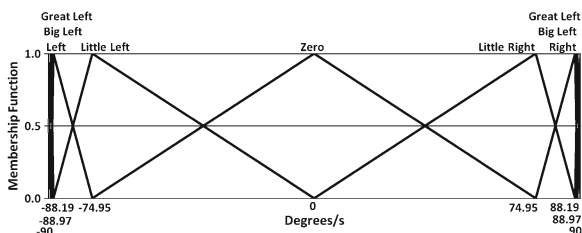

Fig. 29 Membership functions for *Output* (Yaw Command), *after* CE optimization, where, the *Big Left, Left, Little Left* (*Big Right, Right, Little Right*) have been optimized to −88.974, −88.191, −74.952 (88.974, 88.191, 74.952)

Table 7 Rules based on the *Negative* in the third input (integral of the error), *after* CE optimization

Dot error/error	Left	Zero	Right
Big Negative	Left	Little Left	Zero
Negative	Little Left	Zero	Little Right
Zero	Zero	Little Right	Right
Positive	Little Right	Right	Right
Big Positive	Right	Right	Right

between each other, therefore, some sets can be deleted. Two sets of membership functions has been reduced in Figs. 26 and 28, and four sets are deleted in Fig. 29.

These reductions lead to the cancellation of rules directly. Table 6, 7 and 8 shows the final rule base, 64 % of rules has been cancelled from 125 rules to 45 rules, where, Tables 5 (25 rules) and 3 (25 rules) have been cancelled, 10 rules have been reduced in Tables 1, 2 and 4, respectively.

7 Real Fights and Discussions

In the Section 6, we have obtained the optimal SFs and MFs for FLCs, thus, a large number of real flight tests should be done. A quadcopter, i.e. AR.Drone Parrot, is used to test with the FLCs, it connects to the ground station via wireless LAN. In this paper, we use two different types of CE optimized FLCs: (I) only SF is optimized in FLC (called SF-FLC) [33, 35]; (II) both SF and MF are optimized in FLC (named SFMF-FLC). And the flight speeds, i.e. 0.4 m/s, 0.6 m/s and 0.8 m/s, are selected to compare their control performances, the size of Safe Avoiding Area (SAA) is 3 meters. Thus, the collision avoidance process is that the Fail-Safe UAV flies one meter toward the obstacle in Normal Fly Area (NFA), then it should turn 90° in SAA to avoid the collision, the visual

Table 6 Rules based on the *Zero* in the third input (integral of the error), *after* CE optimization

Dot error/error	Left	Zero	Right
Big Negative	Left	Left	Little Left
Negative	Left	Little Left	Zero
Zero	Little Left	Zero	Little Right
Positive	Zero	Little Right	Right
Big Positive	Little Right	Right	Right

Table 8 Rules based on the *Positive* in the third input (integral of the error), *after* CE optimization

Dot error/error	Left	Zero	Right
Big Negative	Left	Left	Left
Negative	Left	Left	Little Left
Zero	Left	Little Left	Zero
Positive	Little Left	Zero	Little Right
Big Positive	Zero	Little Right	Right
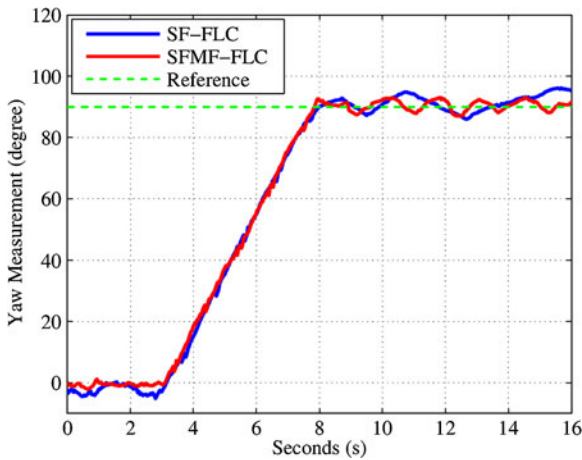
Fig. 30 Measurements of Fail-Safe UAV' heading in the whole collision avoidance task, flight speed is 0.4 m/s

Fig. 32 2D and 3D recontructions for Fail-Safe UAV' trajectories and dynamic changes of heading angle, where, along with Y-axis, NFA: 0–1 m . SAA: 1–4 m. DAA: 4–5 m. Obstacle: 5 m

fuzzy servoing is activated in the whole collision avoidance task. Additionally, we set a threshold, e.g. 5 meters, to the X-axis movements in order to compare all the tests in the same conditions, as shown in the Figs. 32, 35 and 38.

7.1 Test 1: Comparison with Flight Speed: 0.4 m/s

Figure 30 shows the measurements of yaw angle in the whole collision avoidance task with flight speed 0.4 m/s, and Fig. 31 is the enlarged image to show its performance in the steady state. With
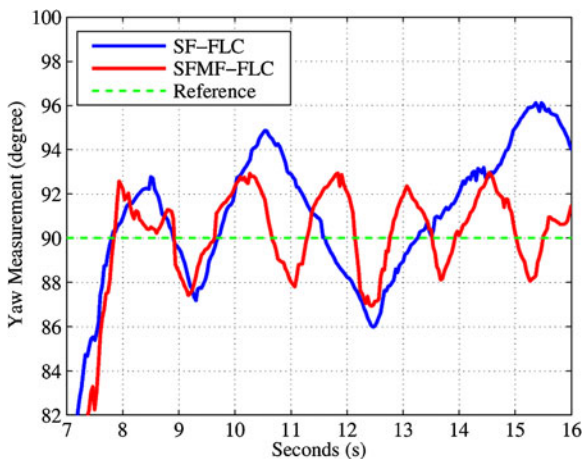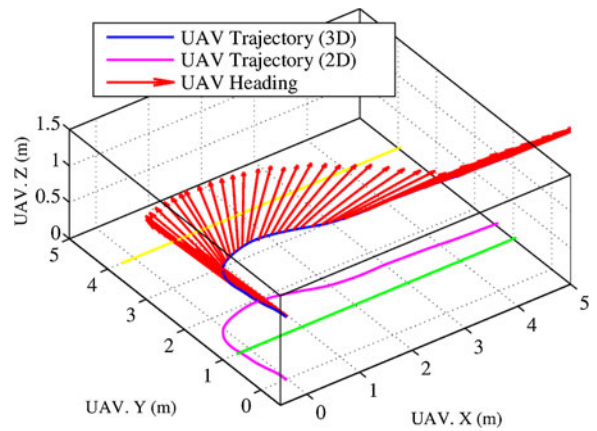
this flight speed, SF-FLC and SFMF-FLC avoided the obstacle successfully and did not fly into the Dangerous Alarm Area (DAA). However, the average RMSE is 4.812° for SF-FLC, while the average RMSE is 2.583° for SFMF-FLC in all the tests. The SFMF-FLC's control performance is better than the one in SF-FLC.

Figure 32 shows the 2D and 3D reconstructions of trajectories and the corresponding dynamic heading angles for Fail-Safe UAV using the SFMF-FLC.
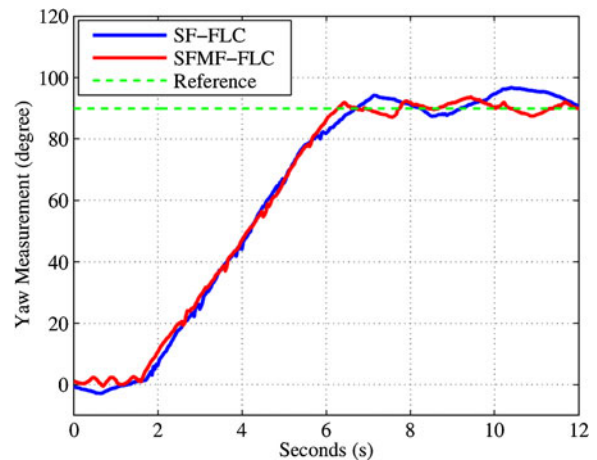

Fig. 31 Enlarged image for steady state performances

Fig. 33 Measurements of Fail-Safe UAV' heading in whole collision avoidance task, flight speed is 0.6 m/s
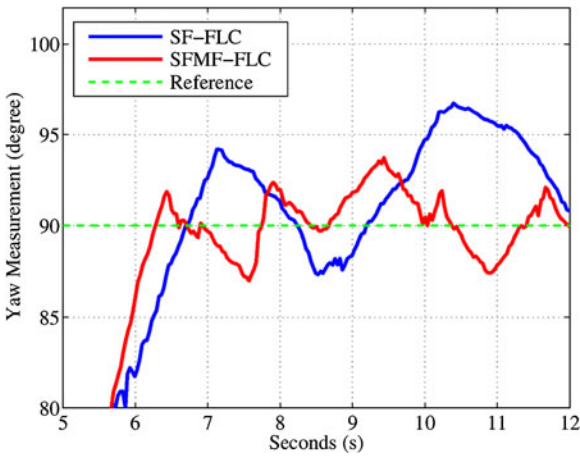
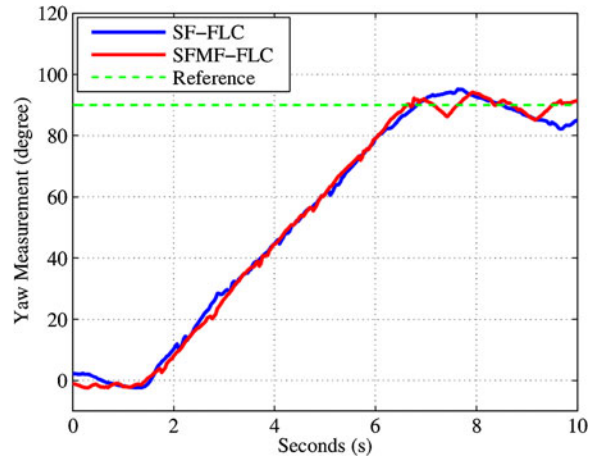**Fig. 34** Enlarged image for steady state performances

Fig. 36 Measurements of Fail-Safe UAV' heading in the whole collision avoidance task, flight speed is 0.8 m/s

7.2 Test 2: Comparison with Flight Speed: 0.6 m/s

Figure 33 shows the heading of Fail-Safe UAV in the whole see-and-avoid task with flight speed 0.6 m/s, and Fig. 34 is the enlarged image to show its performance in steady state. Similarily, with this flight, both controllers also avoided the obstacle successfully and did not fly into the DAA. However, the average RMSE is 6.060° for SF-FLC, while the average RMSE is 3.218° for SFMF-FLC in all the tests. For the control performances, the SFMF-FLC outperforms the SF-FLC.

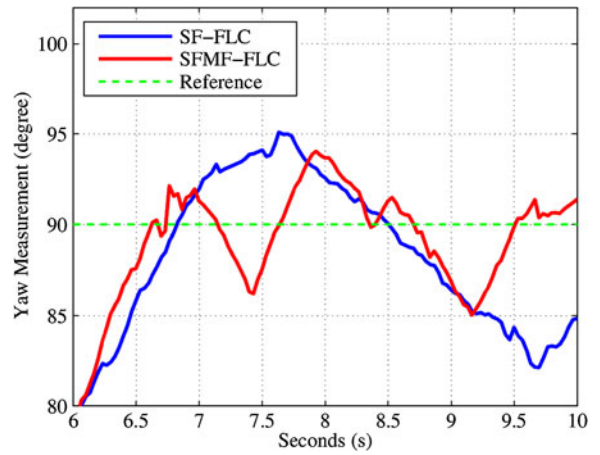Figure 35 shows the 2D and 3D reconstructions of trajectories and the corresponding dynamic

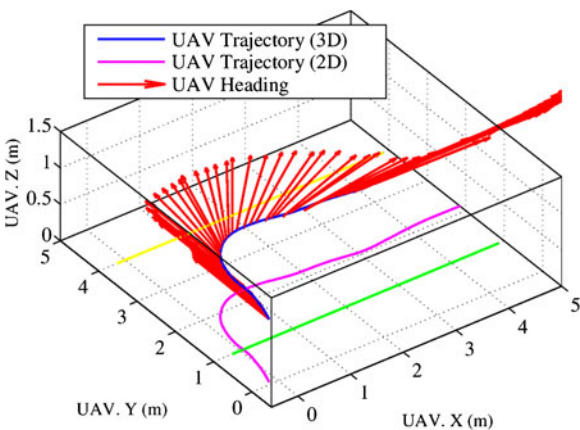**Fig. 37** Enlarged image for steady state performances

Fig. 35 2D and 3D recontructions for Fail-Safe UAV' trajectories and dynamic changes of heading angle
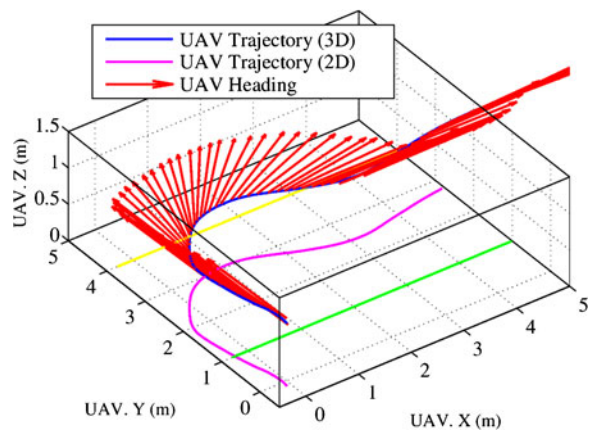

Fig. 38 2D and 3D recontructions for Fail-Safe UAV' trajectories and dynamic changes of heading angle
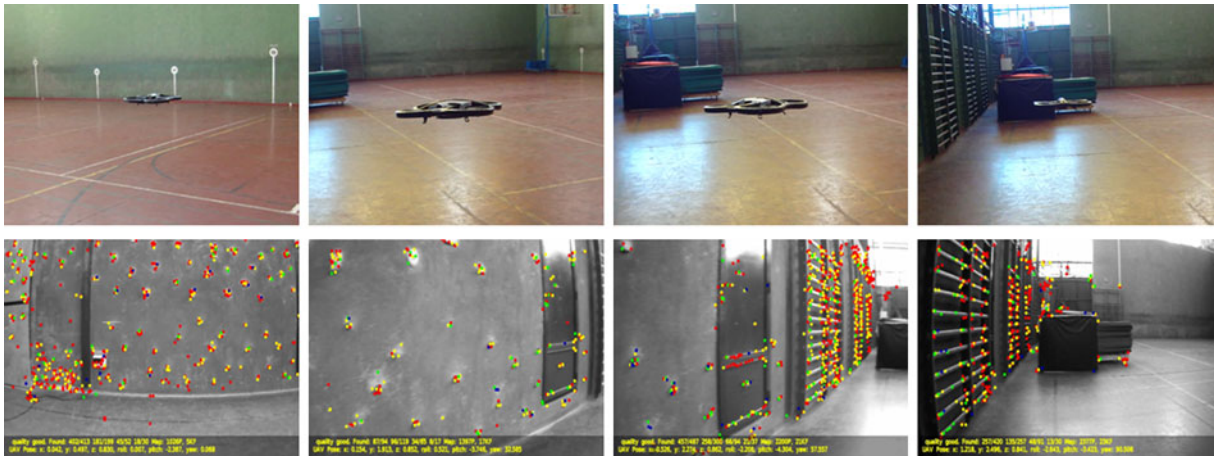
Fig. 39 Fail-Safe UAV in collision avoidance task, where, the *first column*: forward flight (yaw estimation: 0.068°), the *second column*: avoiding with little turning (yaw estimation: 32.585°), the *third column*: avoiding with big turning (yaw estimation: 57.557°) and the *last column*: finish the see-and-avoid task (yaw estimation: 90.506°)

heading angles for Fail-Safe UAV using the SFMF-FLC.

7.3 Test 3: Comparison with Flight Speed: 0.8 m/s

Figure 36 shows the Fail-Safe UAV's yaw angle in the whole collision avoidance task with flight speed 0.8 m/s, and Fig. 37 is the enlarged image to show the performance in steady state. Similarly, both controllers avoided the obstacle successfully and did not fly into the dangerous alarm area. However, the average RMSE is 7.427° for SF-FLC, while the average RMSE is 4.069° for SFMF-FLC in all the tests. For the control performances, the SFMF-FLC is superior to the SF-FLC.

Figure 38 shows the 2D and 3D reconstructions of trajectories and the corresponding dynamic heading angles for Fail-Safe UAV using the SFMF-FLC.

In general terms, all the tests show that by adopting a SFMF-FLC, the Fail-Safe UAV can obtain the better control performances.

Figure 39 shows the external images and real-time processing images in the tests. The related videos and more information of these tests can be found at the CVG-UPM and ColibriProjectUAV[9] websites.

---

8 Conclusions and Future Works

In this paper, the Scaling Factor (SF) and Membership Function (MF) of a Fuzzy Logic Controller (FLC) are optimized by Cross-Entropy (CE) for Fail-Safe UAV, which obtains the better control performance compare to only SF optimization-based FLC [33, 35], therefore, this new FLC is more suitable for collision avoidance applications. And this novel CE optimization not just improves the behavior of FLC, but also reduces 64 % of the initial rule base.

The monocular visual-inertial SLAM-based strategy provides the real-time and accurate localization information for Fail-Safe UAV to control and navigate. Compared to other approaches, this strategy do not need external aids, such as artificial markers, user-specified points/circles, and much computation cost.

For future works, as the field study [36] recommended, the omnidirectional sensor capabilities are needed for obstacle avoidance. For camera, the fisheye lense will be used. In addition, with the development of camera, those cameras with High Dynamic Range (HDR) functions are more promising.

Finally, the optimization to rule weights of FLC will be done, which will casue Microscopic effects to the behavior of FLC to further improve the control peformances.

References

1. Abdelkrim, N., Aouf, N., Tsourdos, A., White, B.: Robust nonlinear filtering for ins/gps uav localization. In: 2008 16th Mediterranean Conference on Control and Automation, pp. 695–702 (2008)

2. Albaker, B.M., Rahim, N.: A survey of collision avoidance approaches for unmanned aerial vehicles. In: 2009 International Conference for Technical Postgraduates (TECHPOS), pp. 1–7 (2009)

3. Angelov, P.: Sense and Avoid in UAS: Research and Applications. Wiley, New York (2012)

4. Babaei, A.R., Mortazavi, M., Moradi, M.H.: Classic and fuzzy-genetic autopilot design for unmanned aerial vehicle. Appl. Soft Comput. **11**, 365–372 (2011)

5. Bonak, M., Matko, D., Blai, S.: Quadrocopter hovering using position-estimation information from inertial sensors and a high-delay video system. J. Intell. Robot. Syst. **67**(1), 43–60 (2012)

6. Caballero, F., Merino, L., Ferruz, J., Ollero, A.: Vision-based odometry and slam for medium and high altitude flying uavs. J. Intell. Robot. Syst. **54**(1–3), 137–161 (2009)

7. Coza, C., Macnab, C.J.B.: A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization. In: Fuzzy Information Processing Society, 2006. NAFIPS 2006. Annual Meeting of the North American, pp. 454–458 (2006)

8. Doitsidis, L., Valavanis, K., Tsourveloudis, N., Kontitsis, M.: A framework for fuzzy logic based uav navigation and control. In: 2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04, vol. 4, pp. 4041–4046 (2004)

9. Dryanovski, I., Valenti, R., Xiao, J.: An open-source navigation system for micro aerial vehicles. Auton. Robot. **34**(3), 177–188 (2013)

10. Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping: part i. IEEE Robot. Autom. Mag. **13**(2), 99–110 (2006)

11. Haber, R.E., del Toro, R.M., Gajate, A.: Optimal fuzzy control system using the Cross-Entropy method. A case study of a drilling process. Inf. Sci. **180**, 2777–279 (2010)

12. Engel, J., Sturm, J., Cremers, D.: Accurate figure flying with a quadrocopter using onboard visual and inertial sensing. In: Proc. of the Workshop on Visual Control of Mobile Robots (ViCoMoR) at the IEEE/RJS International Conference on Intelligent Robot Systems (IROS) (2012)

13. Eresen, A., Imamoglu, N., Efe, M.O.: Autonomous quadrotor flight with vision-based obstacle avoidance in virtual environment. Expert Syst. Appl. **39**(1), 894–905 (2012)

14. He, Z., Iyer, R., Chandler, P.: Vision-based uav flight control and obstacle avoidance. In: American Control Conference, pp. 14–16 (2006)

15. Hrabar, S.: Reactive obstacle avoidance for rotorcraft uavs. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4967–4974 (2011)

16. Kadmiry, B., Driankov, D.: A fuzzy gain-scheduler for the attitude control of an unmanned helicopter. IEEE Trans. Fuzzy Syst. **12**(4), 502–515 (2004)

17. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07) (2007)

18. Zemalache, K.M., Maaref, H.: Controlling a drone: comparison between a based model method and a fuzzy inference system. Appl. Soft Comput. **9**, 404–418 (2009)

19. Kurnaz, S., Cetin, O., Kaynak, O.: Fuzzy logic based approach to design of flight control and navigation tasks for autonomous unmanned aerial vehicles. J. Intell. Robot. Syst. **54**(1–3), 229–244 (2009)

20. Kurnaz, S., Cetin, O., Kaynak, O.: Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles. Expert Syst. Appl. **37**, 1229–1234 (2010)

21. Lenz, I., Gemici, M., Saxena, A.: Low-power parallel algorithms for single image based obstacle avoidance in aerial robots. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 772–779 (2012)

22. Limnaios, G., Tsourveloudis, N.: Fuzzy logic controller for a mini coaxial indoorhelicopter. J. Intell. Robot. Syst. **65**(1–4), 187–201 (2012)

23. Mejias, L., Campoy, I.: Omnidirectional bearing-only see-and-avoid for small aerial robots. In: 2011 5th International Conference on Automation, Robotics and Applications (ICARA), pp. 23–28 (2011)

24. Mellinger, D., Michael, N., Kumar, V.: Trajectory generation and control for precise aggressive maneuvers with quadrotors. Int. J. Robot. Res. **31**(5), 664–674 (2012)

25. Merino, L., Caballero, F., Martnez-de Dios, J., Maza, I., Ollero, A.: An unmanned aircraft system for automatic forest fire monitoring and measurement. J. Intell. Robot. Syst. **65**(1–4), 533–548 (2012)

26. Merz, T., Kendoul, F.: Beyond visual range obstacle avoidance and infrastructure inspection by an autonomous helicopter. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4953–4960 (2011)

27. Michael, N., Mellinger, D., Lindsey, Q., Kumar, V.: The grasp multiple micro-uav testbed. IEEE Robot. Autom. Mag. **17**(3), 56–65 (2010)

28. Murphy, R., Steimle, E., Hall, M., Lindemuth, M., Trejo, D., Hurlebaus, S., Medina-Cetina, Z.,

Slocum, D.: Robot-assisted bridge inspection. J. Intell. Robot. Syst. **64**(1), 77–95 (2011)

29. Nemra, A., Aouf, N.: Robust airborne 3d visual simultaneous localization and mapping with observability and consistency analysis. J. Intell. Robot. Syst. **55**(4–5), 345–376 (2009)

30. Novak, D., Cermak, P.: Advantages of using multi-agent principles in FAIL-SAFE uav system design. In: 2011 16th International Conference on Methods and Models in Automation and Robotics (MMAR), pp. 162–167 (2011)

31. Nutzi, G., Weiss, S., Scaramuzza, D., Siegwart, R.: Fusion of imu and vision for absolute scale estimation in monocular slam. J. Intell. Robot. Syst. **61**(1–4), 287–299 (2011)

32. Olivares-Mendez, M., Campoy, P., Martinez, C., Mondragon, I.: A pan-tilt camera fuzzy vision controller on an unmanned aerial vehicle. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009, pp. 2879–2884 (2009)

33. Olivares-Mendez, M., Mejias, L., Campoy, P., Mellado-Bataller, I.: Cross-Entropy optimization for scaling factors of a fuzzy controller: a see-and-avoid approach for unmanned aerial systems. J. Intell. Robot. Syst. **69**, 189–205 (2013)

34. Olivares-Mendez, M., Mondragon, I., Campoy, P., Martinez, C.: Fuzzy controller for uav-landing task using 3d-position visual estimation. In: 2010 IEEE International Conference on Fuzzy Systems (FUZZ), pp. 1–8 (2010)

35. Olivares-Mendez, M.A., Mejias, L., Campoy, P., Mellado-Bataller, I., Mondragon, I.: Uas see-and-avoid using two different approaches of fuzzy control. In: 2012 International Conference on Unmanned Aircraft Systems (ICUAS'12) (2012)

36. Pratt, K.S., Murphy, R., Stover, S., Griffin, C.: CONOPS and autonomy recommendations for vtol small unmanned aerial system based on hurricane katrina operations. J. Field Robot. **26**(8), 636–650 (2009)

37. Prazenica, R., Kurdila, A., Sharpley, R., Evers, J.: Vision-based geometry estimation and receding horizon path planning for uavs operating in urban environments. In: American Control Conference, 2006, pp. 2874–2879 (2006)

38. Recchia, G., Fasano, G., Accardo, D., Moccia, A., Paparone, L.: An optical flow based electro-optical see-and-avoid system for uavs. In: Aerospace Conference, 2007 IEEE, pp. 1–9 (2007)

39. Ritz, R., Mueller, M., D'Andrea, R.: Cooperative quadrocopter ball throwing and catching. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4972–4978. IEEE (2012)

40. Rubinstein, R.Y., Kroese, D.P.: The Cross-Entropy Method: A Unified Approach to Combinational Optimization, Monte-Carlo Simulation, and Machine Learning. Springer-Berlin, Germany (2004)

41. Santos, M., Lopez, V., Morata, F.: Intelligent fuzzy controller of a quadrotor. In: 2010 International Conference on Intelligent Systems and Knowledge Engineering (ISKE), pp. 141–146 (2010)

42. Scaramuzza, D., Fraundorfer, F.: Visual odometry [tutorial]. IEEE Robot. Autom. Mag. **18**(4), 80–92 (2011)

43. Sturm, J., Bylow, E., Kahl, F., Cremers, D.: Dense tracking and mapping with a quadrocopter. In: Unmanned Aerial Vehicle in Geomatics (UAV-g), Tokyo, Japan (2013)

44. Weiss, S., Scaramuzza, D., Siegwart, R.: Monocular-slam-based navigation for autonomous micro helicopters in gps-denied environments. J. Field Robot. **28**(6), 854–874 (2011)

45. Zadeh, L.A.: Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans. Syst. Man Cybern. **3**(1), 28–44 (1973)

46. Botev, Z., Kroese, D.P.: Global likelihood optimization via the Cross-Entropy method with an application to mixture models. In: Proceedings of the 36th Conference on Winter Simulation (WSC2004), pp. 529–535 (2004)

47. Zheng, L.: A practical guide to tune of proportional and integral (PI) like fuzzy controllers. In: IEEE International Conference on Fuzzy Systems, 1992, pp. 633–640 (1992)

48. Zufferey, J.C., Floreano, D.: Toward 30-gram autonomous indoor aircraft: vision-based obstacle avoidance and altitude control. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005, pp. 2594–2599 (2005)