

Chapter 1

On Using Fuzzy Logic to Control a Simulated Hexacopter Carrying an Attached Pendulum

Emanoel Koslosky, Marco A. Wehrmeister, João A. Fabro
and André S. de Oliveira

Fuzzy logic is used in many applications from industrial process control to automotive applications, including consumers trend forecast, aircraft maneuvering control and others. Considering the increased interest in using of multi-rotor aircrafts (usually called drones) for many kinds of applications, it is important to study new methods to improve multi-rotor maneuverability while controlling its stability in a proper way. Controlling the flight of multi-rotors, specially those equipped six rotors, is not a trivial task. When considering the design of such a control systems, traditional approaches such as PD/PID are very difficult to design, in spite of being easily implementable. This work proposes an approach based on multiple interconnected fuzzy controllers, aiming to control the various aspects related to maneuverability of a hexacopter carrying a free payload forming a pendulum. The behavior produced by such a control system has been simulated on a well-known robotics simulation environment and analyzed in terms of flight stability, as well as roll, pitch and yaw movements. The results show the feasibility of the proposed approach in keeping the hexacopter flying in a stable way.

E. Koslosky (✉) · M.A. Wehrmeister · J.A. Fabro · A.S. de Oliveira
Federal University of Technology - Paraná (UTFPR), Curitiba, Brazil
e-mail: ekosky@gmail.com

M.A. Wehrmeister
e-mail: wehrmeister@utfpr.edu.br

J.A. Fabro
e-mail: fabro@utfpr.edu.br

A.S. de Oliveira
e-mail: andreoliveira@utfpr.edu.br

1.1 Introduction

Nowadays, technology advances and cost reductions have popularized the use of small electromechanical aircrafts in many distinct application fields, such as video recording, plantation inspections, search-and-rescue assistance, military and civil surveillance applications, among others. Multi-rotor helicopters (also known as drones) are among the popular small electromechanical aircrafts that are being used in such applications. Moreover, some of these new applications demand multi-rotor helicopters that fly autonomously, as presented in [4, 7]. Thus, the multi-rotor helicopter must have additional computational systems on top of the more basic movement and stabilization control systems. These computational systems provide higher level capabilities to support the mission accomplishment. Therefore, Unmanned Aerial Vehicles (UAV) are the preferred choice for these applications, due to the cost reductions obtained from eliminating the need of high-skilled and trained pilots.

There are several topologies for multi-rotor helicopter, varying on the number of rotors (i.e. motor and propeller), as well as on the position of these rotors onto the aircraft frame. The most common multi-rotor helicopter has 4 rotors and is called quadcopter. However, recently, other multi-rotor helicopter topologies are becoming popular, such as those with 6 rotors, the so-called hexacopter, as discussed in [9].

The UAV stabilization is commonly performed by hybrid control approaches (parallel, cascade) with multiple PID controllers, like works of [1, 2]. However, these methods require a precise mathematical formulation or identification of UAV dynamics to minimize the disturbance and stabilize the system, as discussed in [11].

Adaptive algorithms can be applied to establish multivariable systems (like UAVs) with more efficiency which classical strategies. In [5] is discusses a approach based on artificial neural networks to trajectory control of UAVs. In [6] the genetic algorithm is applied to establish a hexacopter. In [3] a fuzzy logic method is used to position control of a hexacopter. However, the main focus of previous works is the UAV stabilization over linear disturbances and is not evaluate the proposed control strategies over nonlinear disturbances, like a variable payload.

This work focuses on the control system for the movement and stabilization of a Hexacopter, whose rotors have been configured as a “Hexa +” topology. A multi-layer controller has been proposed and integrates multiple fuzzy controllers. The outputs from these fuzzy controllers must be applied on each rotor accordingly, in order to get the correct Hexacopter movements. A closed control loop is obtained by reading of sensors that measure the position and the movements of the Hexacopter, which, on the other hand, are used as feedback information to the proposed multi-layer controller. The main goal is to create a robust and flexible controller that is able to keep the Hexacopter stability when moving or hovering, even when it carries a free or loose payload that changes its center of gravity.

The major challenge tackled in this work is the switching among fuzzy controllers at the right moment. For instance, lets assume that a hexacopter starts on the ground and receives a command to fly to a certain position, e.g. 2 m in latitude, 5 m in longitude and 3 m upward. To achieve the commanded position, every movement

must be executed properly, and hence, all controllers must work cooperatively to achieve the goal. In other words, the fuzzy controller that controls the longitudinal position cannot override the other controllers actions. On the other hand, when the fuzzy controllers do not work well together, a controller outputs can take the rotors actuation over the outputs of the other controllers. In this case, there must exist a high-level controller that performs the contention of misbehaved controllers.

The proposed approach has been validated through simulation. For that, a model of a hexacopter has been created in the V-REP simulation environment. A free payload has been attached to the hexacopter forming a pendulum. Thus, the proposed multi-layer controller must control the hexacopter movement when it is commanded to move to another position, while keeping the its body stabilized during the flight. Results indicate that the proposed approach is robust since it allows the hexacopter move from one position to another, even though it must carry a moving payload.

The remainder of this chapter is organized as follows: Sect. 1.2 provides an overview of the control problem; Sect. 1.3 presents the proposed stabilization and movement multi-layer fuzzy controller; Sect. 1.4 provides details on each fuzzy controller that comprises the proposed controller; Sect. 1.5 discusses the conducted experiments and the obtained results; finally, Sect. 1.6 draws some conclusions and presents future work directions.

1.2 Description of the Controlled Plant: The Hexacopter

This section describes briefly the system under control, i.e. the hexacopter, as a *control plant*. Figure 1.1 shows the hexacopter, which is composed by six rotors organized as “Hexa +” topology.¹

By activating the rotors accordingly, it is possible to control the hexacopter maneuvering through the X, Y and Z axes.

In general, each plant must be analyzed to discover the interaction of each force. In this case, the thrust force produced by speeding up or slowing down some rotors leads the hexacopter toward the desired direction (on each axis), i.e. the vectors of the forces acting on the plant.

To understand the movements performed by the hexacopter it is worth to take a look at the forces acting in frame.

In the Figs. 1.2 and 1.3 show the forces the rotors imposed to the frame. If these forces are unbalanced the hexacopter start a rotation around the Y-axis and therefore this rotation makes the hexacopter to start a movement over the X-axis. The hexacopter moves forward (Fig. 1.2) if the rear rotor has a value greater than the front rotor. If the front rotor has a value greater than the rear rotor, the hexacopter moves backward or, if it is going forward in this situation, this rotation makes it to slows down.

¹See *APM:Coter – Connect ESCs and Motors*, <http://copter.ardupilot.com/wiki/connect-escs-and-motors/>

Fig. 1.1 The plant to be controlled: (i) X axis angle is the Roll rotation, (ii) the Y axis is the Pitch rotation, and (iii) the Z axis is the Yaw rotation. The *arrow* direction means positive

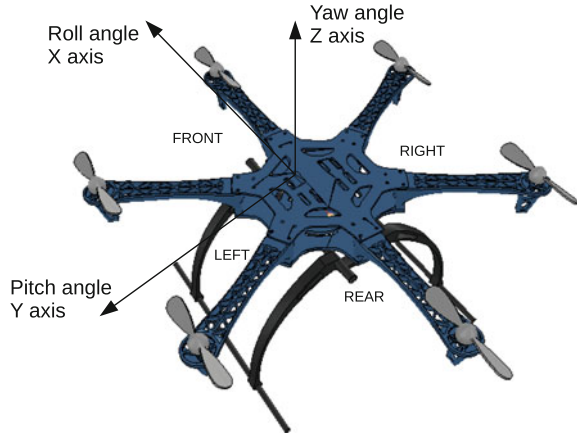


Fig. 1.2 The rear rotor has greater value than front rotor, the hexacopter moves forward

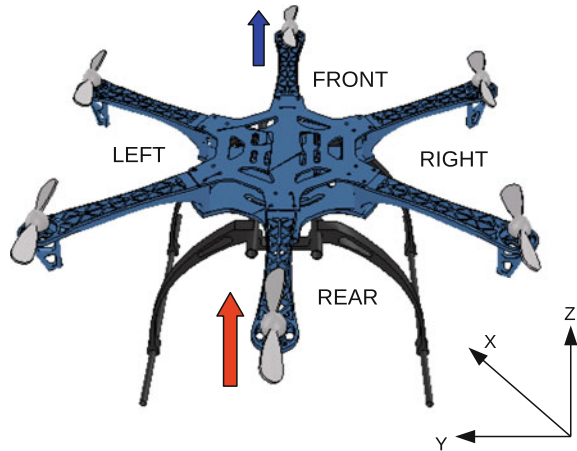


Fig. 1.3 The front rotor has greater value than rear rotor, the hexacopter moves backward

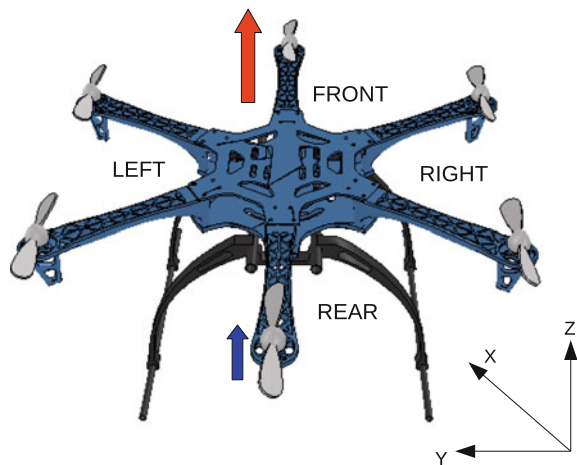


Fig. 1.4 The *left side* rotors have value greater than applied on the *right side* rotors, the hexacopter moves to the *right*, or slows down if it is moving to the *left*

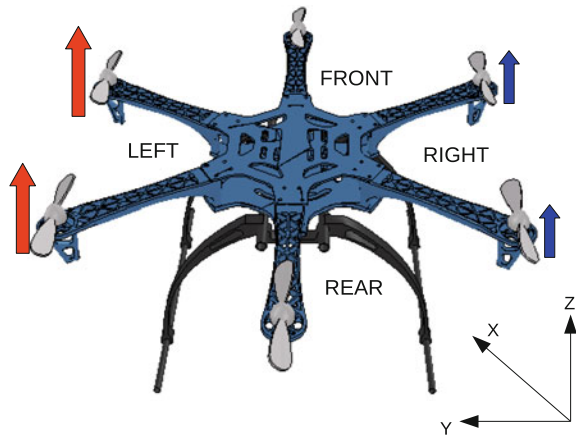
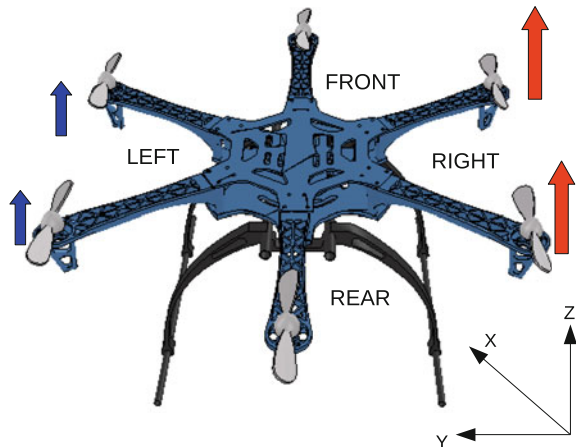


Fig. 1.5 The *right side* rotors have value greater than applied on the *left side* rotors, the hexacopter moves to the *left*, or slows down if it is moving to the *right*



The maneuvers to the right and to the left are achieved by applying forces on the side rotors with different proportion. Thus, it makes the hexacopter rotate around the X-axis and a movement over the Y-axis occurs. For moving to the right or to slow down the movement to the left, the left side rotors have value greater than applied on the right side rotors, as shown the Fig. 1.4. For moving to left or to slow down the movement to right, these forces are inversely applied between the left and right side rotors as shown in the Fig. 1.5.

To rotate the hexacopter in the Z-axis (the yaw movement) the forces are applied alternating among the rotors, shown in Figs. 1.6 and 1.7. It is important to note that if a rotor is set to rotate clockwise, therefore, the adjacent rotor is set to rotate counterclockwise. The real propellers are built with clockwise twist and counterclockwise. If the forces are applied with some difference between the adjacent rotors, a gyroscopic effect begin to act on the frame. This effect is used to rotate the hexacopter around the Z-axis.

Fig. 1.6 The different forces applied to the adjacent rotors to rotate the hexacopter clockwise

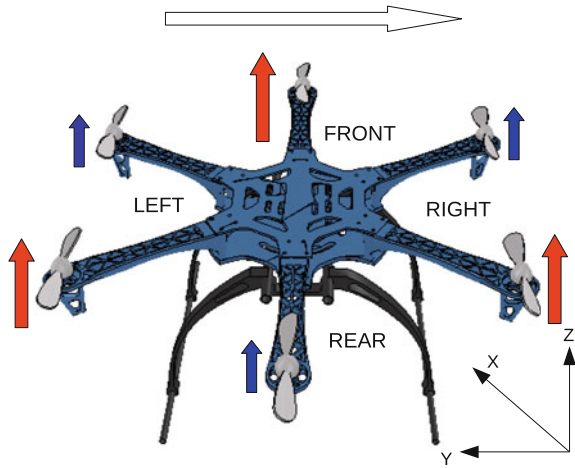
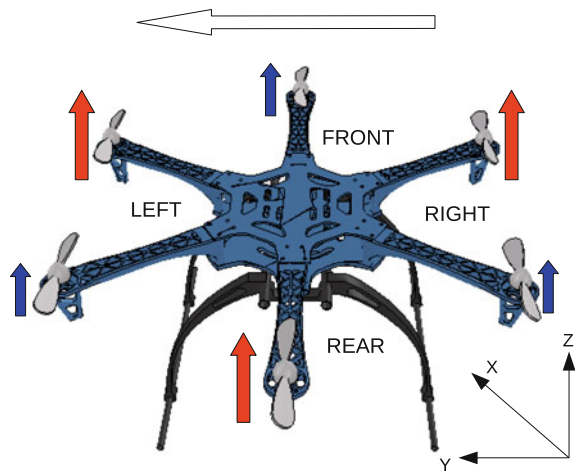


Fig. 1.7 The different forces applied to the adjacent rotors to rotate the hexacopter counter-clockwise



Finally, If all rotors receive the same force, the hexacopter could lift off, land or keep hovering, depending of the intensity. The Figs. 1.8 and 1.9 shows this effect. The hexacopter goes up when the force is high and it goes down when the force is low.

1.3 Proposed Multi-layer Controller

The proposed controller implements a closed loop that comprises the three layers. Data produced as output in one layer is passed as input to the next layer.

Fig. 1.8 With high force, the hexacopter goes up

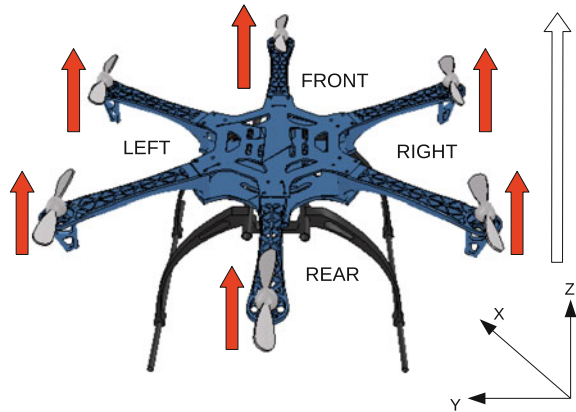
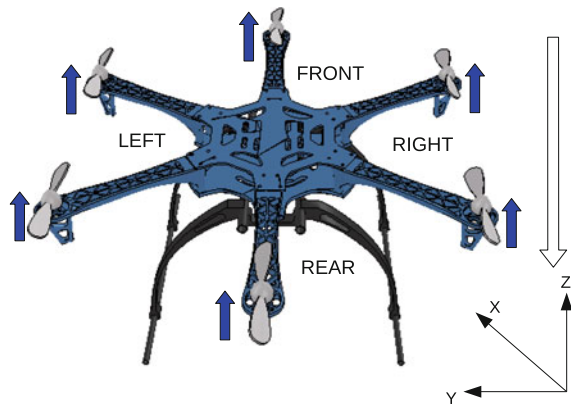


Fig. 1.9 With low force, the hexacopter goes down



The proposed multi-layer fuzzy controller is based on [12] and is depicted in Fig. 1.10. The *Control box* is composed by a pre-processing phase (first layer), a set of fuzzy controllers (second layer), and post-processing phase (third layer).

As one can observe, after the post-processing phase, the control outputs are applied onto the plant by means of the hexacopter rotors that actuate on the hexacopter movement and stabilization. The sensors perceive the changes on the plant controlled variables, and hence, provide the feedback to the controller. The controller, in turn, compares these input values with the reference values established as setpoints thereby closing the control loop [8].

The pre-processing phase (first layer) is responsible for acquiring data from the input sensors, process the input movement commands, as well as calculate the controlled data used as input to the fuzzy controllers in second layer. Before the multi-layer controller starts its execution, there is an initialization phase that is performed within the first layer. The target position is set as the current position, so that the hexacopter does not move before receiving any command. Gyroscope and accelerometer sensors are calibrated and the GPS sensor is initialized by gathering at least four

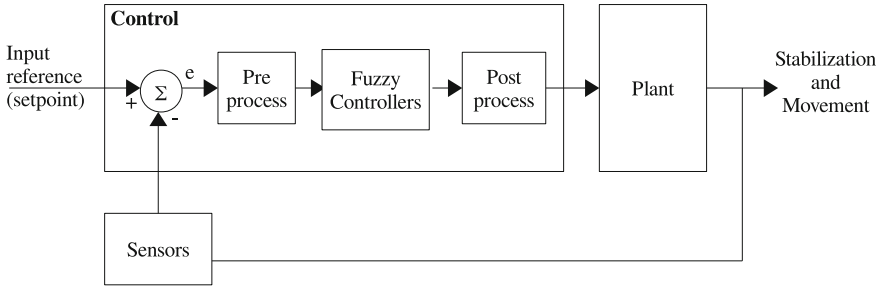


Fig. 1.10 Inside the control box. Composed by preprocess, fuzzy controllers and postprocess

satellites. During the execution phase, the first layer is responsible to calculate the input variables to the fuzzy controllers: (i) the angular and linear distance (delta error) for X, Y, and Z axes between the current hexacopter position and the target position; (ii) the rotation and translation movement matrices to translate 3 axes movement into the speed related to the ground (i.e. X and Y axis). In addition, it is responsible to convert the input movement commands into setpoints for X, Y and Z positions. Movements commands are composed of three values representing the positive or negative movement along X, Y and Z axes related to the current positions, i.e. a command indicates a relative position. Thus, when a new command is received, the first layer will convert it to an absolute position. Then, when the control system is executing, this layer uses the GPS coordinates to determine the error in the distance from the hexacopter to the target position. These calculated errors in position are the inputs to the fuzzy controllers (Euler X, Euler Y and Euler Z errors).

The second layer contains five fuzzy controllers, which act on issues regarding the hexacopter movement, namely hovering stabilization, vertical and horizontal movement and heading. As mentioned, these controllers take as input the data produced in the first layer and generate output for the third layer. The generated outputs represent the actuation on the six rotors for performing pitch, roll, yaw moves for all maneuvers necessary to reach the target position. The fuzzy controllers are discussed in details in the next section.

The post-processing phase (third layer) is responsible for coordinating the fuzzy controllers outputs. As mentioned, in order to perform a proper maneuver, the proposed multi-layer controller establishes a priority on movements needed to complete a maneuver. When a new command is received, i.e. a new target point is set, the hexacopter must firstly reach the target altitude. Then, the hexacopter must turn until its front aims the target position. Finally, the hexacopter moves horizontally towards the target position. This layer also performs a threshold limits control by means of output values saturation, in order to keep the hexacopter stability while flying or hovering.

1.3.1 Pre-processing Phase

In the pre-processing phase some calculations are applied onto the fuzzy controllers. Each fuzzy controller (see Sect. 1.4) has at least two inputs [12]: the error $e(t)$ and some derivative variable such as speed.

The error is calculated according to Eq. 1.1. In other words, $e(t)$ is the difference between the reference value $r(t)$ and the current value of an sensor $y(t)$.

The reference is a setpoint established by an operator or other controller. Some inputs have the reference always set to zero so that the error $e(t)$ is the opposite value of sensor, i.e. $-y(t)$.

$$e(t) = r(t) - y(t) \quad (1.1)$$

The errors calculated in the pre-processing phase are related to the following variables. The rotation around the three axes is the Euler angle (in radians) and it is measured by a gyroscope. The following outputs are generated:

- Angle on X axis: Roll_error
- Angle on Y axis: Pitch_error
- Angle on Z axis: Yaw_error

The *distance* to the target position (i.e. the error between the current hexacopter position and the setpoint on the three axes) is measured in meters using GPS and calculated as described bellow.

The following outputs are generated:

- Horizontal distance on X axis: distX_error
- Horizontal distance on Y axis: distY_error
- Vertical distance on Z axis: distZ_error

The distance ds is calculated to determine the distance to the target position that is decomposed by X and Y axes, as shown in Eqs. 1.2 and 1.3.

Then, the Euclidean Distance is calculated to obtain the real distance d to the target (see Eq. 1.4). Euclidean Distance d is also used in Eq. 1.6 to calculate the speed $v(t)$.

$$\Delta ds_x = s_{x_t} - s_{x_{(t-1)}} \quad (1.2)$$

$$\Delta ds_y = s_{y_t} - s_{y_{(t-1)}} \quad (1.3)$$

$$\Delta d = \sqrt{\Delta d_x^2 + \Delta d_y^2} \quad (1.4)$$

Moreover, the angle of the movement is determined by the arctangent as shown in Eq. 1.5. The angle $a(t)$ is used as the new yaw setpoint.

$$a(t) = \arctangent\left(\frac{\Delta d_x}{\Delta d_y}\right) \quad (1.5)$$

The *linear speed* is measured in meters per second and is calculated according to Eq. 1.6 as discussed below. The following outputs are generated:

- Horizontal speed on X axis: SpeedX_error
- Horizontal speed on Y axis: SpeedY_error
- Vertical speed on Z axis: SpeedZ_error

The distance Δs and the time interval Δt are used to determine the current speed of the hexacopter.

The time interval Δt is obtained by measuring the time instant on which two consecutive values of Euclidian Distance are calculated.

$$v(t) = \frac{\Delta d}{\Delta t} \quad (1.6)$$

Despite of acceleration is a derivative of speed, such a variable is taken directly from accelerometer sensor. The acceleration information is used together other others measurements like Euler angles in order to avoid oscillation movements. For instance, the acceleration measured over Y-axis is used together Euler angle error measured around the X-axis. If the Euler angle error is zero, it means the hexacopter is stabilized accordingly to the X-angle. But it could be moving over the in Y-axis like drifting. The controller must to slow down this movement. If not stop this drifting, a oscillation begin appear. In order to the fuzzy controller realize this moment, the acceleration measurement is used as input together with Euler angle to be processed by controller. The controllers are described in detail in the Sect. 1.4.

Another calculation present in pre-processing phase is the rotation matrix [15]. It is used to obtain the speed and distance error over the X and Y axes related the pose of hexacopter and the inertial frame. In other words, the information from GPS tells the hexacopter position on the world (the inertial frame) but nothing about the pose of it. By applying the rotation matrix calculation, using the Euler Z-angle error, it is possible to discover the pose of hexacopter on the world as well as the speed related to its X and Y axis. With this information the controller can determine the correct forces to be applied onto each rotor.

1.3.2 Post-processing Phase

The post-processing phase determines the movement sequencing, saturation and so on. The output of fuzzy controllers are provided as the inputs to this phase. Such a behavior is explained as the Finite State Machine (FSM) depicted in Fig. 1.11.

There are many approaches for controlling the movement of a hexacopter. For instance, one could create an algorithm in which the hexacopter flies directly to the target position in the three-dimensional space by changing the vehicle altitude and horizontal position at same time.

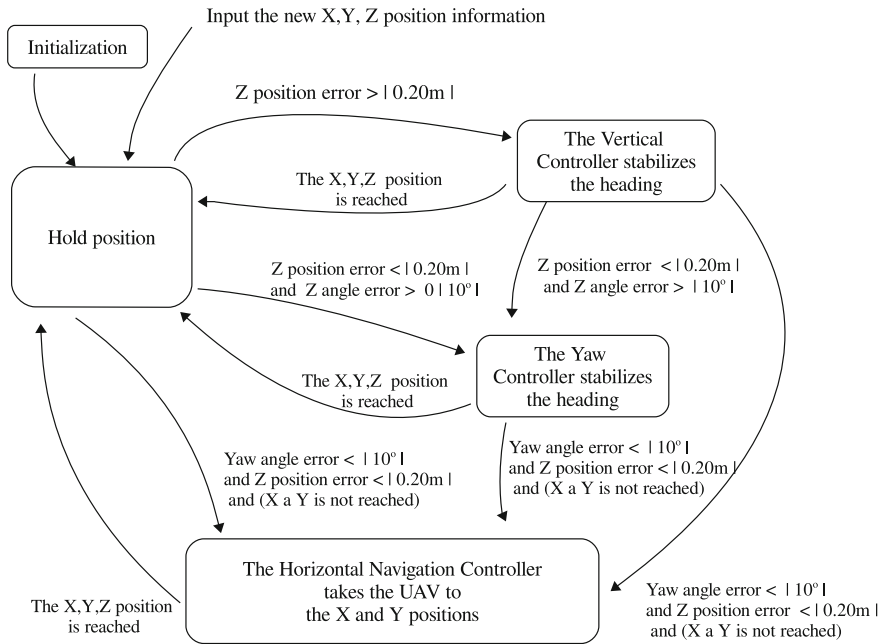


Fig. 1.11 FSM for sequencing the hexacopter maneuver process. (X, Y, Z) inputs represent the new target position of the hexacopter

Due to the instability caused by the loose payload, the controller proposed in this work adopts an approach that sequentializes the flying movements. As presented in Fig. 1.11, the first movement of hexacopter is to reach the desired altitude (Z-axis position). Thereafter, the controller commands the hexacopter to aim directly at the target position by rotating on the Y-axis. Finally, the controller commands the hexacopter to move along with X and Y axes, in order to reach the target position.

After the initialization, the systems goes to the *Hold Position* state. In this state, *Roll*, *Pitch* and *Hovering* movements are stabilized by their fuzzy controllers. Once a new command is provided by the operator, it changes the X, Y and Z position setpoints, leading to an increase on error values as described in Sect. 1.3.1. The proper fuzzy controller is activated when the threshold of one of its input values is reached. For instance, *Horizontal Navigation* fuzzy controller starts when the altitude and yaw is under a certain threshold. Such thresholds controls the hexacopter stabilization. It is important to note that if some external disturbance interfere with the hexacopter stabilization, the controller stops the horizontal movement until the input values reach their thresholds.

The proposed multi-layer controller implements the saturation control in the post-processing phase, in order to avoid an individual fuzzy controller to override other fuzzy controllers outputs by means of dominating the actuation on the plant.

Therefore, once the output of logic control is calculated, it is important to determine the exact value that must be applied on each rotor.

When throttle value is the same to all rotors, the hexacopter keeps hovering on the same position. On the other hand, the output from roll and pitch stabilization are applied proportionally as a gain to the rotors throttle according to the Eqs. 1.7–1.12.

$$\text{PropellerForceFRONT} = \text{Othrottle} - \text{Othrottle} \times \text{Opitch}; \quad (1.7)$$

$$\text{PropellerForceRIGHT_FRONT} = \text{Othrottle} - \text{Othrottle} \times (\text{Oroll}/2); \quad (1.8)$$

$$\text{PropellerForceRIGHT_REAR} = \text{Othrottle} - \text{Othrottle} \times (\text{Oroll}/2); \quad (1.9)$$

$$\text{PropellerForceREAR} = \text{Othrottle} + \text{Othrottle} \times \text{Opitch}; \quad (1.10)$$

$$\text{PropellerForceLEFT_REAR} = \text{Othrottle} + \text{Othrottle} \times (\text{Oroll}/2); \quad (1.11)$$

$$\text{PropellerForceLEFT_FRONT} = \text{Othrottle} + \text{Othrottle} \times (\text{Oroll}/2); \quad (1.12)$$

In these equations, $\text{PropellerForce}\langle\text{Rotor Position}\rangle$ is the value applied on the rotor, the Othrottle is the output from *Hovering fuzzy controller*, and the Opitch and Oroll are the outputs from, respectively, *Pitch* and *Roll fuzzy controllers*. It is worth mentioning that: (i) to maintain the opposite feedback into the mesh, the output value is obtained by subtracting Oroll from the throttle value for rotors at the right side of the hexacopter, as well as by subtracting Opitch for the front rotor. Similarly, for the left-side and rear rotors, respectively, Oroll and Opitch are added to the throttle value; and (ii) Oroll values are proportional to the amount of rotors on the right/left sides of the hexacopter, i.e. Oroll value is divided by two. Such proportional values avoid that right and left Othrottle values do not override the front and rear Othrottle values.

1.4 The Fuzzy Controllers

1.4.1 The Fuzzy Method

A fuzzy controller can be created with a variety of types of membership functions such as trapezoidal, triangle, Gaussian bell curve function, and others. In addition, these function may be of receive many inputs and provide a simple output (MISO) or receive many inputs and provide many outputs (MIMO).

The fuzzy controller proposed in this work are composed five independent fuzzy controllers. These controllers are built from MISO membership functions defined as trapezoidal and triangle forms. The $\min()$ operator has been used in the rule inferences and the result is done by $\max()$ operator.

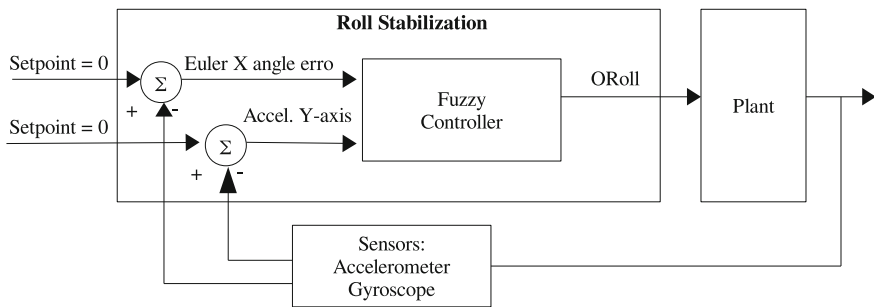


Fig. 1.12 The roll stabilization fuzzy controller

Moreover, the defuzzification method used in this work is Center of Gravity (COG). The next sections provides details on these five independent fuzzy controller.

1.4.2 Roll Stabilization

Roll is the movement obtained through the rotation around the X-axis, i.e. front-to-back axis. The fuzzy controller named Roll Stabilization controls the stabilization of the hexacopter while it is performing the roll maneuver. Figure 1.12 shows the block diagram of this controller.

This controller has two input data. The first input is error in roll angle Euler approximation. The roll angle is calculated through the Euler approximation of the current X angle and the target X-axis angle. Figure 1.13 shows the linguistic variable membership function representing the fuzzification of the error in the roll angle Euler approximation. The second input is the perceived movement in Y-axis represented as the acceleration in Y-axis obtained from the accelerometer over the time. Figure 1.14 shows the linguistic variables and the membership functions representing

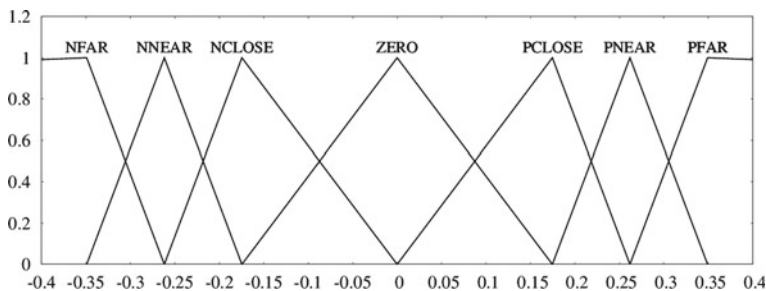


Fig. 1.13 Input linguistic variables and their membership functions for the roll angle, Euler approximation error of X-axis angle

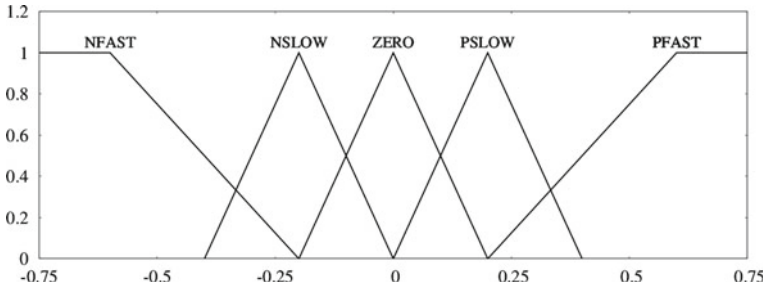


Fig. 1.14 Input linguistic variables and their membership functions for: Y-axis accelerations

Table 1.1 Control rules for roll stabilization

Roll angle	NFAST	NNEAR	NCLOSE	ZERO	PCLOSE	PNEAR	PFAST
AccelY							
NFAST	PMAX	PMIN	ZERO	PMIN	NMIN	NMID	NMAX
NSLOW	PMAX	PMIN	ZERO	ZERO	NMIN	NMID	NMAX
ZERO	PMAX	PMID	PMIN	ZERO	NMIN	NMID	NMAX
PSLOW	PMAX	PMID	PMIN	ZERO	ZERO	NMIN	NMAX
PFAST	PMAX	PMID	PMIN	NMIN	ZERO	NMIN	NMAX

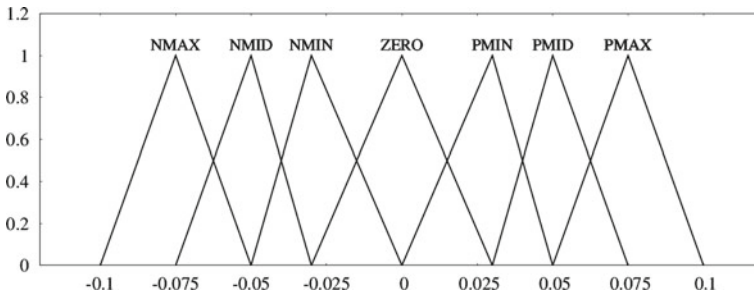


Fig. 1.15 Output linguistic variables and their membership functions for: ORoll

the fuzzification of Y-axis acceleration. Its worth mentioning that the “N” and “P” prefixes of variables names stand for, respectively, Negative and Positive.

The roll stabilization fuzzy controller is composed of 35 rules as show in Table 1.1. The output of this controller is the omega roll variable (ORoll), whose values are depicted in Fig. 1.15. The defuzzification of ORoll variable creates the values that control the rotation speed of right- and left-hand side rotors, which, in turn, produce enough force to make the hexacopter rotate in the X-axis. It is important to highlight that X-axis and Y-axis acceleration and also roll and pitch angle error variables are used to minimize (or correct) the stabilization interference caused by pendulum effect created by the free payload.

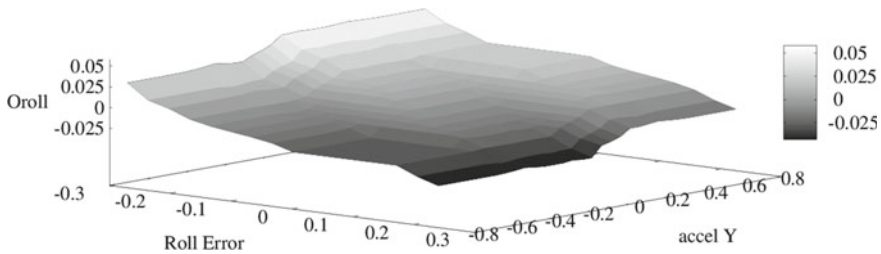


Fig. 1.16 Fuzzy control surface for roll stability control

The fuzzy control surface graphic is shown in Fig. 1.16; it depicts the relationship among inputs and output. It is possible to note that at near the point zero of roll error the output is smooth. When the the roll error is far from zero the correction is greater. Also we can see that acceleration over the Y-axis in all the surface plays a important role: it avoids the oscillation. For instance, suppose the error is about 0.1, a rotation to the right makes the hexacopter to fly to the right direction. When the hexacopter is rotating and the angle error is reaching zero, the hexacopter starts to fly faster. Without considering the information on acceleration, the hexacopter tends to oscillate. Another way to see this effect is the following: the acceleration over Y axis is negative when roll error is positive and vice-versa. However, there are situations in which the hexacopter moves in positive direction in respect to the Y-axis, the roll error is positive. This means that the hexacopter must start to fly slower and the angle must be kept unchanged or raised. Without the acceleration information, the output of this controller could become zero (when roll error is zero) and the hexacopter might drift. Some derivative value in time, e.g. speed or acceleration, helps to avoid oscillation or sliding.

1.4.3 Pitch Stabilization

Pitch is the movement obtained through the rotation around the Y-axis, i.e. side-to-side axis. The pitch stabilization fuzzy controller holds the hexacopter stabilization while it is performing the pitch maneuver. Figure 1.17 shows the block diagram of this controller.

This controller is very similar to the Roll Stabilization controller. It has two input variables: (i) the error in pitch angle calculated through Euler approximations; and (ii) X-axis acceleration to indicate a possible movement in the X-axis. The linguistic variables and the membership function values for these two input variables are depicted in Figs. 1.18 and 1.19, respectively. In the same way, pitch stabilization is also defined by five rules as described in Table 1.2, and has only one output the omega pitch variable (OPitch) depicted in Fig. 1.20. The definition of these variables and rules are exactly the same as in the Roll Stabilization fuzzy controller. However,

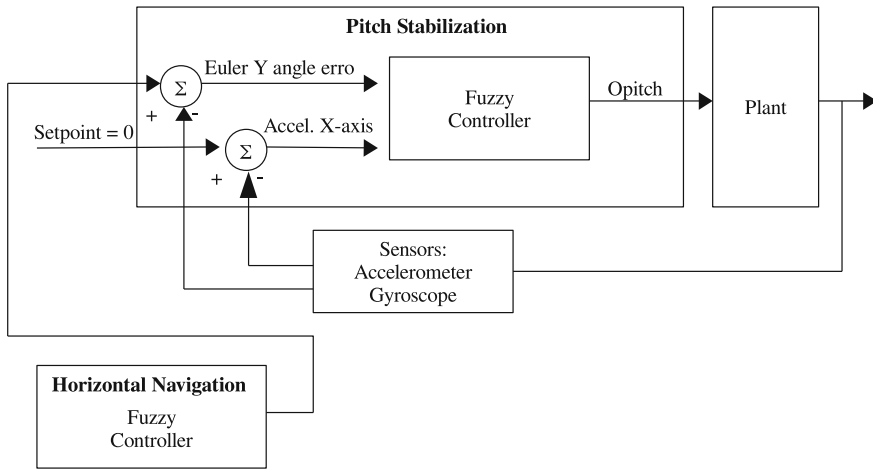


Fig. 1.17 The pitch stabilization fuzzy controller

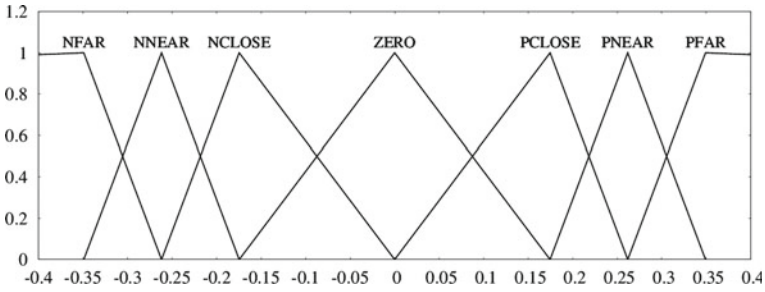


Fig. 1.18 Input linguistic variables and their membership functions for: Pitch angle, Euler approximation error of Y-axis angle

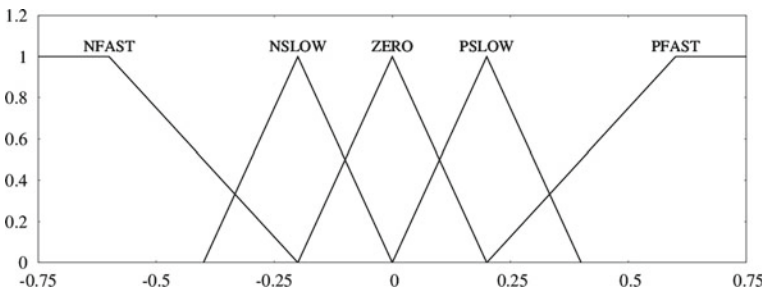


Fig. 1.19 Input linguistic variables and their membership functions for: X-axis

Table 1.2 Control rules for pitch stabilization

Roll angle	NFAR	NNEAR	NCLOSE	ZERO	PCLOSE	PNEAR	PFAR
AccelY							
NFAST	PMAX	PMIN	ZERO	PMIN	NMIN	NMID	NMAX
NSLOW	PMAX	PMIN	ZERO	ZERO	NMIN	NMID	NMAX
ZERO	PMAX	PMID	PMIN	ZERO	NMIN	NMID	NMAX
PSLOW	PMAX	PMID	PMIN	ZERO	ZERO	NMIN	NMAX
PFAST	PMAX	PMID	PMIN	NMIN	ZERO	NMIN	NMAX

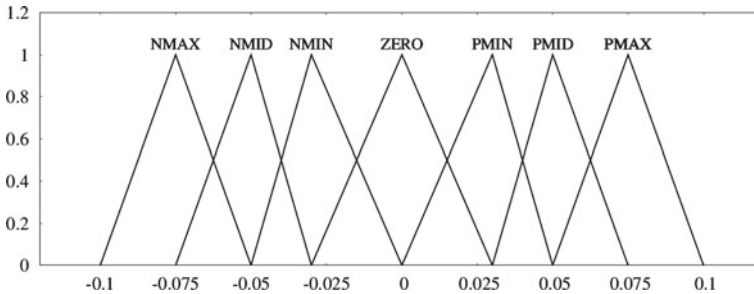


Fig. 1.20 Output linguistic variables and their membership functions for: OPitch

it is important to highlight that the defuzzification process on OPitch generates values that control the speed of the front and rear rotors making the hexacopter rotate on the Y-axis. It is done by Horizontal navigation controller described later.

Similarly, the pendulum effect mentioned previously is minimized or mitigated by means of using the pitch angle error and Y-axis acceleration variables. The roll and pitch stabilization fuzzy controllers are key to maintain the hexacopter stabilized while it is flying or hovering. Equally important is the way these controllers actuate on the rotors. The correction applied to a controlled “defuzzified” variable is proportional to its current value, see Eqs. 1.5–1.11. In other words, instead of simply summing a new absolute actuation value to a variable (e.g. the rotor speed), the gain is proportional to current value. This improves the controller efficiency in extreme situations, e.g. when the corrective value is insignificant (compared to the current value) or the correction value is too high, avoiding aggressive corrections, and hence, improving stability. Finally, it is worth noting that the overall hexacopter stabilization is performed by both roll and pitch stabilization fuzzy controllers.

The surface graphic about pitch stability control depicted in Fig. 1.21 shows the relationship between pitch angle error and the accelerometer information over the X-axis. Both input values work together to stabilize the hexacopter over the Y-axis and avoid oscillation over the X-axis. For instance. Suppose the pitch controller receive the pitch error as input value close to zero and as the acceleration input value high. It means the hexacopter is in movement forward even though the pitch angle error is zero. Thus, the controller must to slow down it. The control surface shows

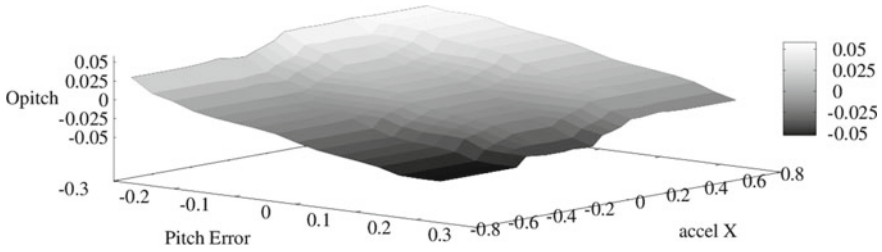


Fig. 1.21 Fuzzy control surface for pitch stability control

that the output pitch angle (Opitch) is greater than zero in this situation, it makes the hexacopter rotate around Y-axis counter-clockwise causing it to slow down.

The forward movement is obtained by changing the Y-axis setpoint computed by *Horizontal navigation controller*.

1.4.4 Heading to Goal (Yaw Controller)

Yaw is the rotation movement around the vertical axis, that is, the Z-axis. The yaw controller (or heading goal controller) is responsible for pointing the hexacopter front to the target position, keeping this position until it arrives at the destination. Figure 1.22 shows the block diagram of this controller.

To allow this controller to perform such a task, two input variables are needed. The yaw angle is similar to roll and pitch angles, and hence, are calculated based on the Euler approximation of Z-axis angle between the current angle position of the hexacopter front and the target position. Figure 1.23 shows the linguistic variable and the membership function values of yaw angle. The distance to the goal is the second input data utilized. It indicates how far the hexacopter from the target position. The

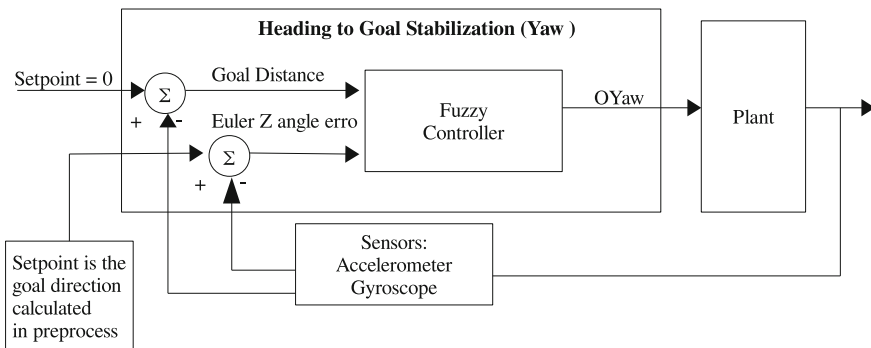


Fig. 1.22 The heading to goal stabilization fuzzy controller (The Yaw Stabilization)

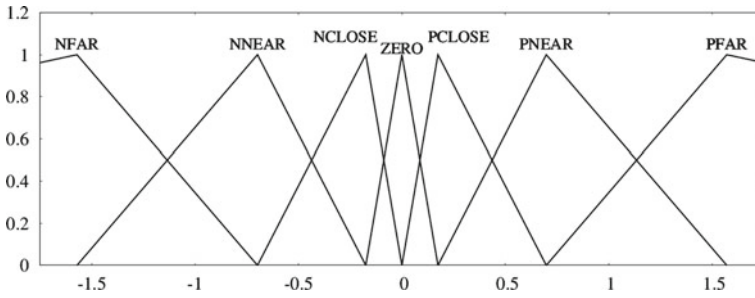


Fig. 1.23 Input linguistic variables and their membership functions for: Yaw angle error

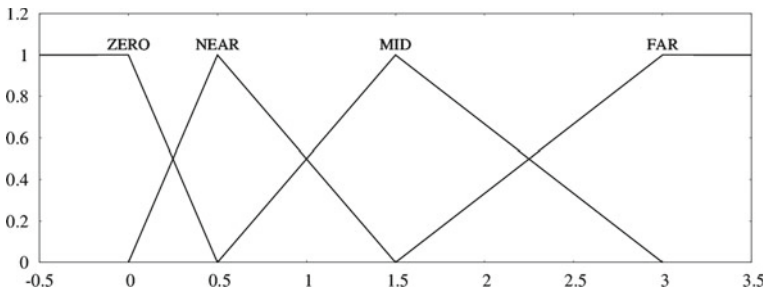


Fig. 1.24 Input linguistic variables and their membership functions for: Goal distance to determines the hexacopter heading

Table 1.3 Yaw control rules

Yaw angle	NFAR	NNEAR	NCLOSE	ZERO	PCLOSE	PNEAR	PFAR
Distance							
ZERO	ZERO	ZERO	ZERO	ZERO	ZERO	ZERO	ZERO
NEAR	PMAX	PMID	PMIN	ZERO	NMIN	NMID	NMAX
MID	PMAX	PMID	PMIN	ZERO	NMIN	NMID	NMAX
FAR	PMAX	PMID	PMIN	ZERO	NMIN	NMID	NMAX

goal distance is calculated from X- and Y-axis positions of the hexacopter and results in a polar coordinate indicating the angle and the distance to the target point. Goal distance linguist variable is depicted in Fig. 1.24.

This fuzzy controller has 28 rules as shown in Table 1.3.

These rules define the value of the output variable omega yaw (OYaw), presented in Fig. 1.25. OYaw is defuzzified and create actuation values that are applied on all rotors or on only a few of them. Depending on which rotor are affected the hexacopter turns clockwise or counter-clockwise. On the other hand, heading goal (yaw) controller gains priority over the other controllers when the hexacopter reaches

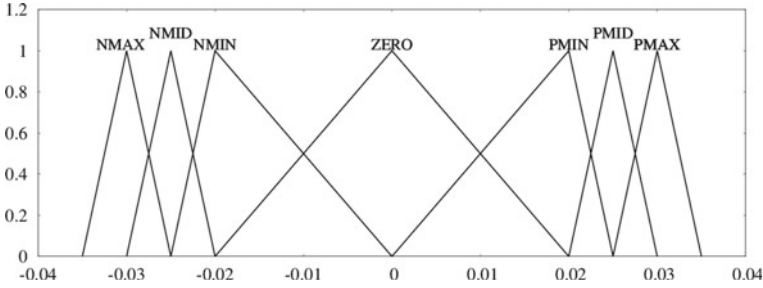


Fig. 1.25 Output linguistic variables and their membership functions for: OYaw

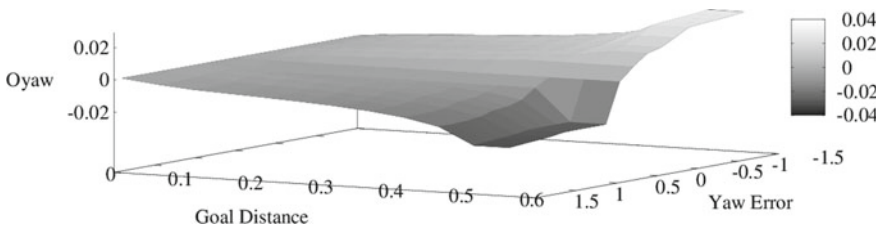


Fig. 1.26 Fuzzy control surface for Yaw stability control

a region within 70 cm radius around the target point. Thus, rover and pitch controllers are only responsible to main the hexacopter hovering stable on the target position.

Figure 1.26 shows the surface graphic of the *Yaw controller*. It works only when the goal distance is greater than the threshold, about 40 cm or higher in the *Goal Distance* axis on the surface graphic. If the hexacopter is on the close to the target, the Yaw controller must be disabled. Otherwise, he stay spinning indefinitely.

1.4.5 The Horizontal Navigation

Horizontal navigation fuzzy controller controls the hexacopter fly on the X- and Y-axis. Figure 1.27 shows the block diagram of this controller. It takes as input the goal distance (see Sect. 1.4.4) and the horizontal speed.

The latter is calculated as traveled distance divided by time, i.e. the difference between the goal distance of two consecutive polar coordinates divided by the time elapsed between their calculation. Figure 1.28 shows the linguistic variable associated with the horizontal speed, while Fig. 1.29 shows the linguistic variable for the goal distance.

The rules of the horizontal navigation fuzzy controller are presented in Table 1.4. This controller results in the pitch angle for navigation (OPitchNavigation) as depict in Fig. 1.30 which, in turn, affects the pitch stabilization fuzzy controller. Specifically, the pitch angle for navigation moves the stability point towards the target

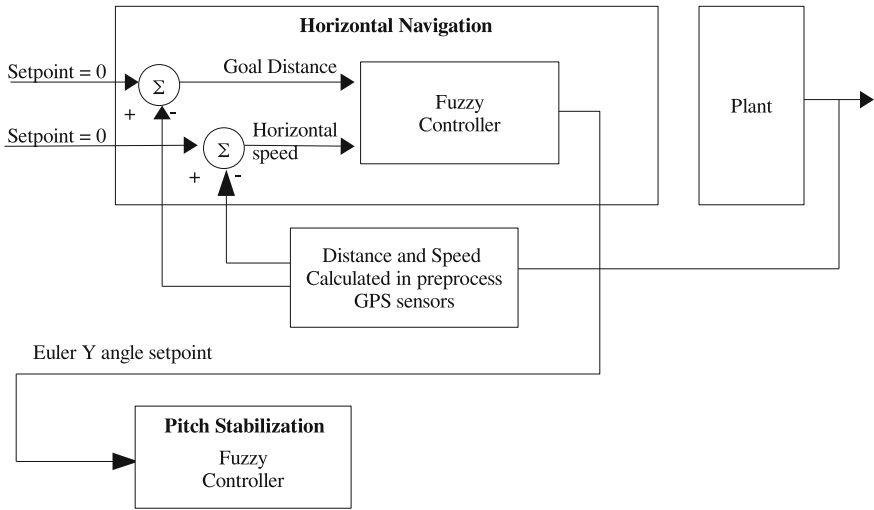


Fig. 1.27 The horizontal navigation fuzzy controller

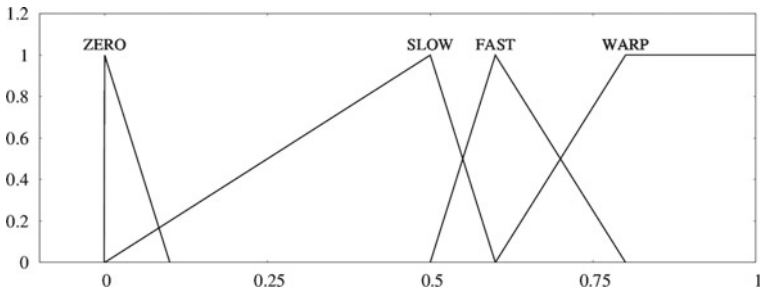


Fig. 1.28 Input linguistic variables and their membership functions for: Horizontal speed

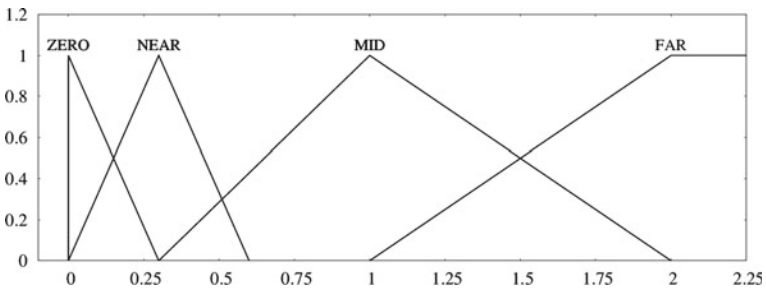


Fig. 1.29 Input linguistic variables and their membership functions for: Goal distance

Table 1.4 Horizontal navigation control rules

Goal distance	ZERO	NEAR	MID	FAR
Horizontal speed				
ZERO	ZERO	ZERO	PMAX	PMAX
SLOW	NMIN	PMIN	PMAX	PMAX
FAST	NMID	PMIN	PMID	PMAX
WARP	NMAX	NMIN	PMIN	PMAX

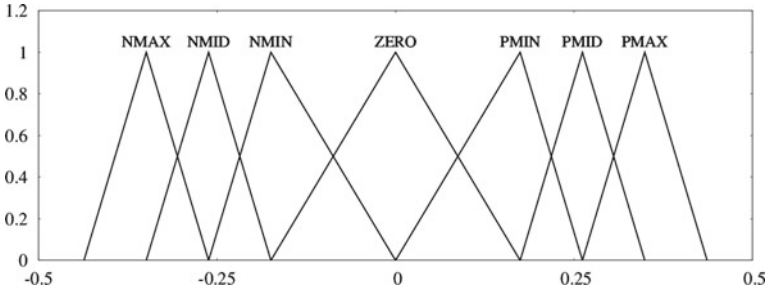


Fig. 1.30 Output linguistic variables and their membership functions for: OPitchNavigation (the Pitch setpoint)

direction, making a hexacopter to fly forwards. It is important to note that, by using the mentioned inputs, horizontal navigation fuzzy controller commands the horizontal movement in a smooth way, i.e. as the hexacopter comes close to the target point, its horizontal speed decreases in order to alleviate the control bounce produced by the movement inertia.

The horizontal navigation control surface is depicted in Fig. 1.31. When the hexacopter is on the target position, the goal distance is near to zero and the horizontal speed is zero, the pitch setpoint is zero. In this state the pitch controller stabilizes the hexacopter on the current position, and hence, hexacopter is kept with pitch angle at zero. When the goal distance is greater than zero, this controller changes the pitch setpoint.

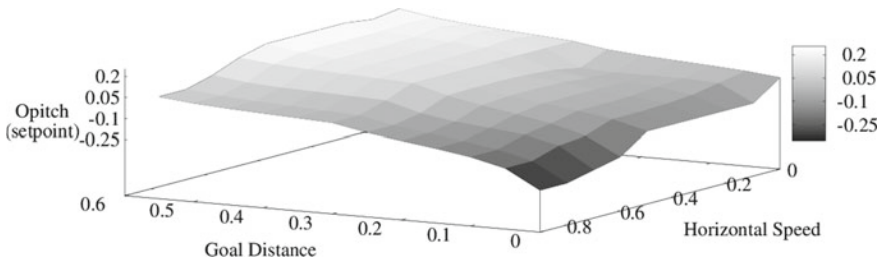


Fig. 1.31 Fuzzy control surface for horizontal navigation control

Thus, the pitch controller stabilizes the hexacopter in new pitch angle, which higher than zero, making the hexacopter to move forward.

When the goal distance is zero and the horizontal speed is high, the output is negative. In this situation the vehicle is moving while passing over the target position, and hence, it must slow down. This is the reason why this controller output must be negative.

1.4.6 The Vertical Navigation

Vertical navigation fuzzy controller is similar to the horizontal navigation controller. However, it controls the movement on the Z-axis. Figure 1.32 shows the block diagram of this controller. This controller takes as input the vertical distance to the target point, as well as the vertical speed.

The first one is the error in the Euler approximation of Z-axis, while the second one is the difference in distance (Z-axis). Linguistic variables for vertical distance and vertical speed are presented in Figs. 1.33 and 1.34, respectively.

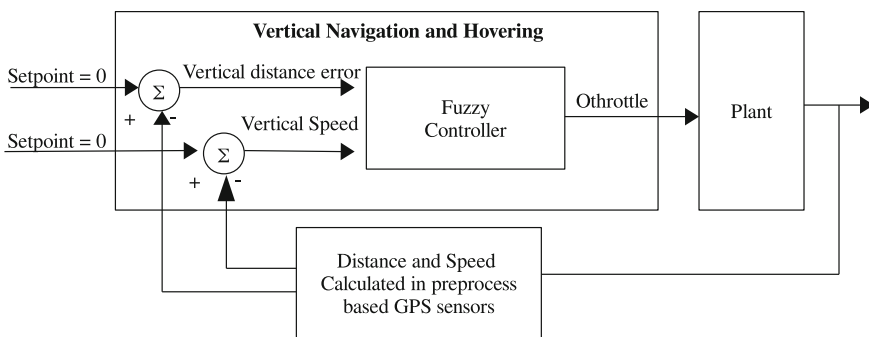


Fig. 1.32 The vertical navigation fuzzy controller

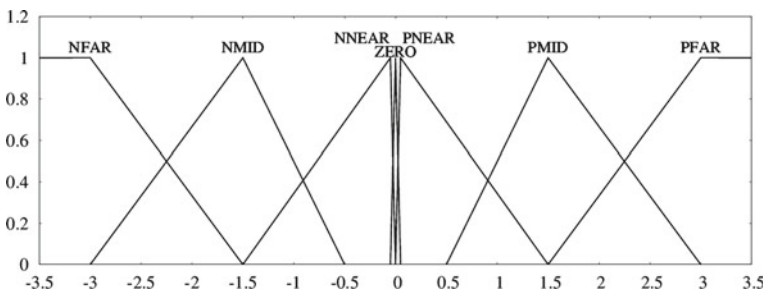


Fig. 1.33 Input linguistic variables and their membership functions for: Vertical distance

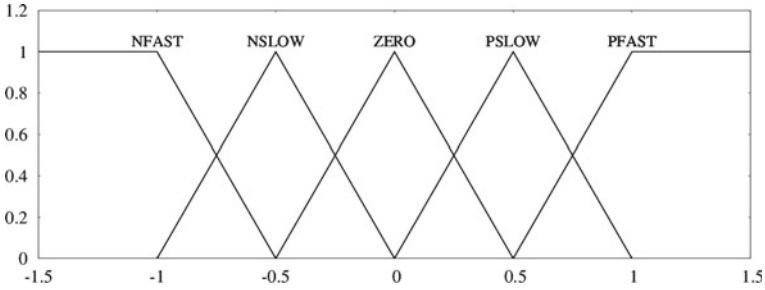


Fig. 1.34 Input linguistic variables and their membership functions for: Vertical speed

Table 1.5 Vertical navigation control rules

V. dist.	NFAST	NMID	NNEAR	ZERO	PNEAR	PMID	PFAST
V. speed							
NFAST	NMAX	NMID	NMIN	PMIN	PMID	PMAX	PMAX
NSLOW	NMAX	NMID	NMIN	PMIN	PMID	PMAX	PMAX
ZERO	NMAX	NMID	NMIN	ZERO	PMIN	PMID	PMAX
PSLOW	NMAX	NMAX	NMID	NMIN	PMIN	PMID	PMAX
PFAST	NMAX	NMAX	NMID	NMIN	PMIN	PMID	PMAX

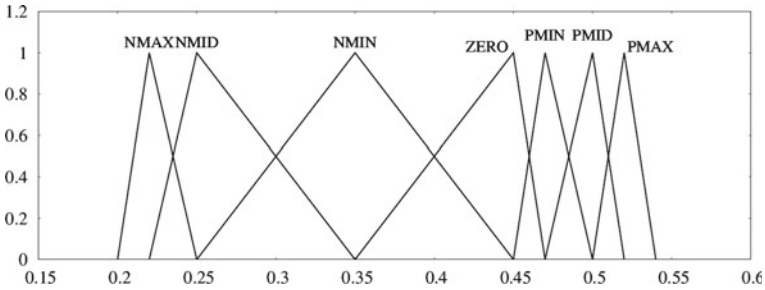


Fig. 1.35 Output linguistic variables and their membership functions for: OThrottle

Table 1.5 shows the 35 rules that compose the vertical navigation fuzzy controller. As the result, this controller sets the omega throttle variable (OThrottle), presented in Fig. 1.35 which is decomposed in the amount of power applied on all rotors, increasing or decreasing the overall lift force making the hexacopter fly on higher or lower altitude. It is worth noting that this presents a smooth control approach similar to the horizontal navigation, i.e. the power applied on the rotors decreases along with vertical speed as the hexacopter comes closer to target altitude.

The fuzzy surface control for vertical navigation and hovering is shown in Fig. 1.36. The altitude is maintained by controlling the throttle applied onto the all rotors. The input information is taken from the GPS sensor. The vertical speed is

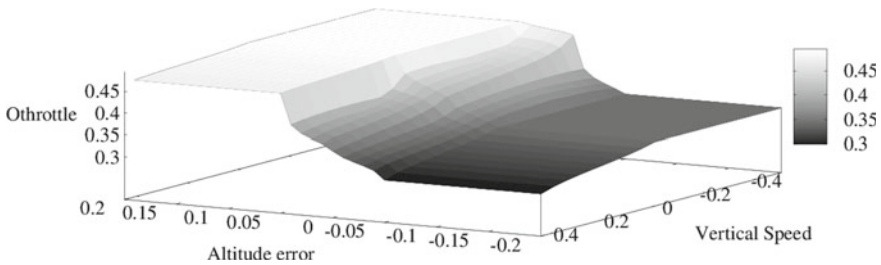


Fig. 1.36 Fuzzy control surface for vertical navigation and hovering control

used to avoid the hexacopter to oscillate up and down. It is similar as the acceleration is used to avoid oscillation in the pitch and roll controller. It is worth to note that the ZERO output not means zero value, but the value that the hexacopter is hovering. To realize the relationship between altitude error and vertical speed, suppose the altitude error is zero, the hexacopter is in the target vertical position, but also suppose the vertical speed is positive, perhaps 0.4 or higher. It means the hexacopter reached the target and goes beyond because it is in movement to up. It must be slowed down. Therefore, the controller sets the output value to a value lower than the ZERO, causing the hexacopter to slow down. On the other hand, when the altitude is zero and the vertical speed is negative it means the hexacopter is falling down. In this condition, the controller must to set output to a value higher than the ZERO just to make the hexacopter stop the falling.

1.5 Experiment and Results

1.5.1 Overview of the Experiment

The work has been validated through a case study by means of simulation. For that, a model of a hexacopter has been created in the V-REP robotics simulation environment. A free payload has been attached to the hexacopter forming a pendulum as depicted in Fig. 1.37. In the simulated environment, the hexacopter weighs 980 g (mass = 0.1) and the payload weighs 49 g (mass = 0.005). The hexacopter model used is one that is already available on V-REP. Such a payload weight was defined in 5% of the hexacopter weight due to limitations on the rotors model that cannot provide enough thrust to allow the hexacopter takeoff. For simulation the V-REP has been configured with “Dynamic engine” as “Bullet”, the “Dynamics settings” as “Verry accurate” and “Simulation time step” as “dt = 10 ms”.

The major challenge tackled in this work is the switching among fuzzy controllers at the right moment. For instance, let us assume that a hexacopter starts on the ground and receives a command to fly to a certain position, e.g. 3 m in latitude, 3 m

Fig. 1.37 Moving carrying a weight



in longitude and 3 m upward. To achieve the commanded position, every movement must be executed properly, and hence, all controllers must work cooperatively to achieve the goal. In other words, the fuzzy controller that controls the longitudinal position cannot override the other controllers actions. On the other hand, when the fuzzy controllers do not work well together, a controller output could take the rotors actuation over the output of the other controller **controllers' outputs might overlap each other**. In this case, there must exist a high-level controller that performs the contention of misbehaved controllers.

The proposed approach has been validated through simulation. For that, a model of a hexacopter has been created in the V-REP simulation environment. A free payload has been attached to the hexacopter forming a pendulum. Thus, the proposed multi-layer controller must control the hexacopter movement when it is commanded to move to another position, while keeping the its body stabilized during the flight. In other words, the results indicate that the proposed approach is robust since it allows the hexacopter move from one position to another, even though it must carry a moving payload.

The proposed multi-layer fuzzy controller has been implemented on Linux using the C language and gcc compiler version 4.9.1. This software communicates with V-REP environment, receiving the input sensor signals and sending the output commands generated by the proposed fuzzy controller. The main idea was to develop hybrid fuzzy controllers. The control system is composed of three threads: (i) a thread for the main control loop; (ii) a thread to produce a data log which was used to create the charts presented in this section; and (iii) another one to insert the user commands as target point coordinates used as setpoints to the the proposed controller. Therefore, the developed software acts as both the hexacopter movement and stability controller and the interface between the operator and the hexacopter.

The experiment has been performed in two phases. The first one has concentrated on calibrating the range of values for all linguistic variables of the five fuzzy controllers. The results of this phase was described in see Sect. 1.4. Those values have

Table 1.6 The position sent to hexacopter as command

Commands	X	Y	Z
1st	+3	+3	+3
2nd	-2	+1	+5
3rd	+4	-2	+8
4th	-2	-1	+2
5th	0	0	+2
6th	0	0	0

been defined in a manual and arbitrary way by means of an iterative trial and error process. In this phase, the aim was to achieve a stabilized movement and hovering for the hexacopter. For that, the hexacopter has flown without attached payload. The hexacopter has been commanded to fly to six distinct positions from the origin point (i.e. $X = 0$, $Y = 0$, $Z = 0$) by means of the following relative coordinates expressed in meters shown in Table 1.6.

The ranges for each value of all linguistic variables have been defined using an intuitive try-and-error method. On each simulation round, threshold limits have been tuned until the hexacopter was able to fly and hover on a fixed point in a stable way.

The second phase focused on evaluating how the proposed multi-level fuzzy controller behaves when the hexacopter carries a free payload. In other words, this phase assesses how the proposed controller behaves in situations of stress caused by the pendulum effect created by the inertial movement of the free payload. The same commands have been issue in the same order as described earlier. All data generated during the simulation have been analyzed and the results are discussed in the next section.

1.5.2 Results

This section presents the results in terms of how the values of the controlled variables evolved during the experiment. As mentioned, once the hexacopter receives the command to fly to a new position, it performs the following sequence of actions: (1) the hexacopter flies up until reaching the target altitude; (2) the controller established the new yaw angle in order to aim at the desired X- and Y-axis position; and, finally, (3) the hexacopter flies horizontally toward the target position. This sequence of steps is obtained by establishing thresholds in the transitions of motion events, as depicted in Fig. 1.11. Figure 1.38 shows the footprint of the trajectory flown by the hexacopter during the experiment.

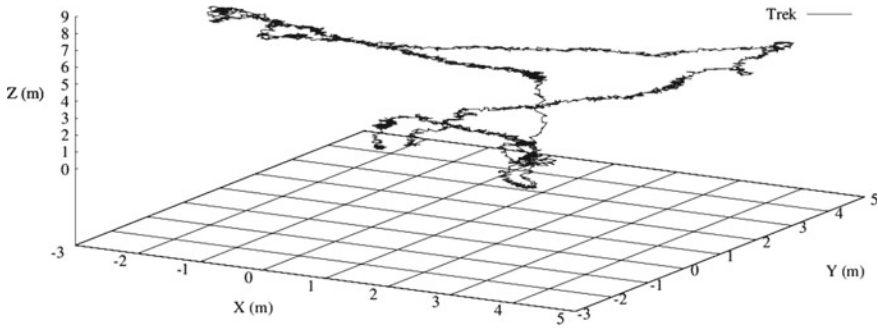


Fig. 1.38 Flight footprint

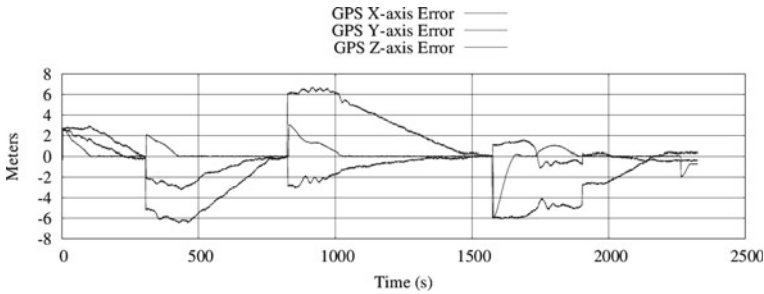


Fig. 1.39 Movement and hovering stabilization control

Figures 1.39 shows the error of the GPS coordinates that have been read during the entire flight. The moment when a new command has been received by the hexacopter, please refer to Table 1.6: (i) the 1st command was received at 0s; (ii) the 2nd command at 300s; (iii) the 3rd command at 800s; (iv) the 4th command at 1600s; (v) the 5th command at 2500s;

Videos of the experiment can be seen on:

<https://www.youtube.com/watch?v=dTJhH8lU6BY>.

Flying data generated during the simulation can be observed on Figs. 1.40, 1.41, 1.42 and 1.43.

1.5.3 Discussion

It is worth noting that, despite some disturbance caused by the definition of a new position, the hexacopter moved smoothly and in a stable way from the current position to the target position. In addition, the proposed multi-layer fuzzy controller has responded properly to the control stress imposed by the free payload. Such a claim is

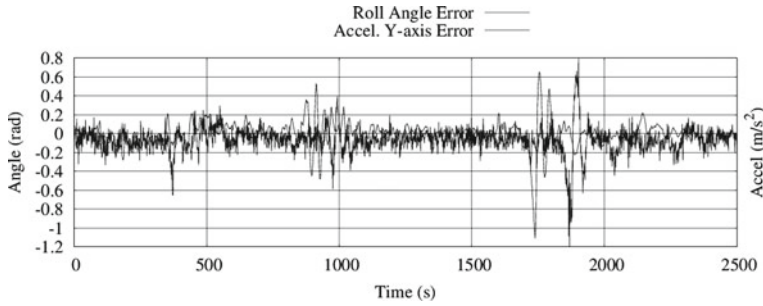


Fig. 1.40 Roll stabilization—inputs

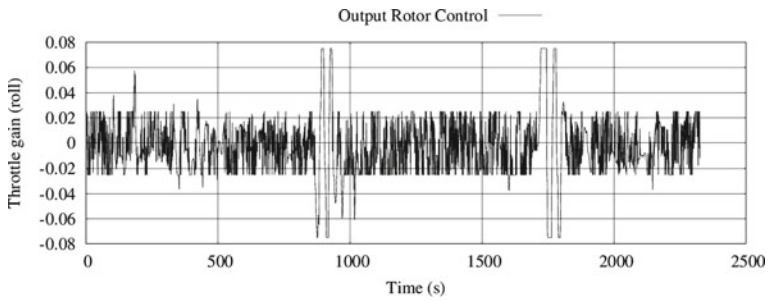


Fig. 1.41 Roll stabilization—output

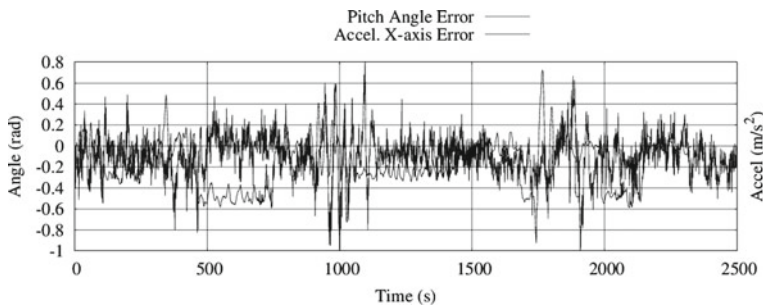


Fig. 1.42 Pitch stabilization—input

supported by the analysis of flying data plotted on the charts presented in Figs. 1.40, 1.41, 1.42, and 1.43.

In these charts, at moments close to 300, 800, 1600 and 2500 s, the present variation seems to indicate a poor stability.

However, at each time instant, a new target position is sent to the hexacopter, disturbing the system. Therefore, the controllers must act to control the hexacopter stability. The output variable value of these controllers varies around ± 0.08 representing 8% variation within 150s, which is considered acceptable for a stable

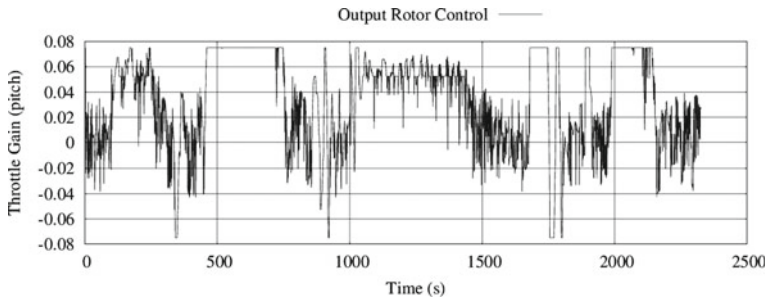


Fig. 1.43 Pitch stabilization—output

flight of a hexacopter carrying a free payload. A video of the simulation shows the hexacopter flying stable, as well as it shows the hexacopter recovering from these disturbances.

Furthermore, tuning membership functions took a considerable time. However, once they are correctly calibrated with the unloaded hexacopter, the proposed fuzzy controller was able to control the movement and the stability without modifications. Thus, it was observed the flexibility of fuzzy logic for designing a complex control systems as the one presented in this work. Other perceived advantage of fuzzy logic is the handling of non-linear scale independently from input or output, for instance the scale for the throttle output, goal distance, pitch navigation output.

During the calibration phase, various models have been tried in order to properly control the hexacopter stability during horizontal and vertical navigation – the stabilized hovering has been achieved easily. It was not enough to sum the value OPitch and ORoll to the throttle variable (OThrottle). It has been observed that V-REP simulates random errors while the simulation is running as it happens in the real world. Such errors affect the throttle proportionally, and hence, to keep the roll and pitch stabilization the outputs of these fuzzy controllers must follow this growth. To cope with this situation, outputs of pitch and roll controls were modified to represent a gain based on a percentage of the current throttle (see Eqs. 1.7–1.12).

1.6 Conclusions and Future Work

This work proposes an approach that integrates several fuzzy controller that work collaboratively to keep the stabilization and control the navigation of an Hexacopter carrying a free payload while it is flying and hovering. This paper discusses how the propose controller has been designed.

In order do evaluate the proposed approach, a simulation experiment has been conducted. The proposed approach was able to stabilize the hovering, control the altitude, position, and navigation of the simulated hexacopter. Additionally, the proposed multi-layer fuzzy controller has responded properly to the control stress imposed by

the free payload. When the hexacopter has flown with the free payload, the control system took more time to stabilize. By using fuzzy logic, it was possible to see the flexibility of the proposed approach, since once the variables have been calibrated, it was not necessary to change the systems to allow the hexacopter to fly stable carrying an attached free payload. However, on the other hand, the side-effect of this flexibility is the difficult to tune the variables thresholds.

The next steps involve the implementation of the proposed multi-layer fuzzy controller on a real Hexacopter. A computing systems with several distinct sensors is envied so that it allows moving towards a fully autonomous UAV that executes localization, map building, path planning and mission execution. In addition, considering the difficult of tuning the proposed systems in the calibration phase, other future work direction is to develop adaptive fuzzy control to assist in this task.

References

1. Ahmed OA, Latief M, Ali MA, Akmeliawati R (2015) Stabilization and control of autonomous hexacopter via visual-servoing and cascaded-proportional and derivative (PD) controllers. 2015 6th international conference on automation, robotics and applications (ICARA), pp 542–549. IEEE
2. Alaimo A, Artale V, Milazzo CLR, Ricciardello A (2014) PID controller applied to hexacopter flight. *J Intell Robot Syst* 73:261–270
3. Bacik J, Perdukova D, Fedor P (2015) Design of fuzzy controller for hexacopter position control. artificial intelligence perspectives and applications. *Advances in intelligent systems and computing*. Springer International Publishing, Berlin, pp 192–202
4. Bipin K, Duggal V, Madhava K (2015) Autonomous navigation of generic monocular quadcopter in natural environment. *Autonomous navigation of generic monocular quadcopter in natural environment*, pp 1063–1070. IEEE
5. Collotta M, Pau G, Caponetto R (2014) A real-time system based on a neural network model to control hexacopter trajectories. 2014 international symposium on power electronics, electrical drives, automation and motion (SPEEDAM)
6. Genetic algorithm applied to the stabilization control of a hexarotor (2015) Genetic algorithm applied to the stabilization control of a hexarotor. In: *Proceedings of the international conference on numerical analysis and applied mathematics*, vol 1648
7. Haque Md R, Muhammad M, Swarnaker D, Arifuzzaman M (2014) Autonomous Quadcopter for product home delivery. 2014 international conference on electrical engineering and information & communication technology (ICEEICT), pp 1–5. IEEE
8. Lee EA, Seshia SA (2015) *Introduction to embedded systems – a cyber-physical systems approach*, 2nd edn. Springer, New York www.leeSeshia.org, Chapters 1–3
9. Leishman R, Macdonald J, McLain T, Beard R (2012) Relative navigation and control of a hexacopter. *Autonomous Quadcopter for product home delivery*. 2012 IEEE international conference on robotics and automation (ICRA)
10. Ma sum MA, Jati G, Arrofi MK, Wibowo A, Mursanto P, Jatmiko W (2013) Autonomous quadcopter swarm robots for object localization and tracking. 2013 international symposium on micro-nanomechatronics and human science (MHS)
11. Oldziej D, Gosiewski Z (2013) Modelling of Dynamic and control of six-rotor autonomous unmanned aerial vehicle. *solid state phenomena*. Trans Tech Publ, pp 220–225
12. Passino KM, Yurkovich S (1998) *Fuzzy control, 1.2–conventional control system design*. Addison-wesley, Boston Subsection 2.2

13. Rohmer E, Singh SPN, Freese M (2013) V-REP: A versatile and scalable robot simulation framework. 2013 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 1321–1326. IEEE
14. Salih AL, Moghavvemi M, Mohamed HAF, Gaeid KS (2010) Scientific Research and Essays. Flight PID controller design for a UAV quadrotor. Academic Journals, Denmark
15. Siegwart R, Nourbakhsh IR, Scaramuzza D (2011) Introduction to autonomous mobile robots. MIT press, Cambridge