Research Article

Stable Hovering Flight for a Small Unmanned Helicopter Using Fuzzy Control

Arbab Nighat Khizer,^{1,2} Dai Yaping,¹ Syed Amjad Ali,^{2,3} and Xu Xiangyang¹

¹ School of Automation, Beijing Institute of Technology, Beijing 100081, China

² Mehran University of Engineering and Technology, Jamshoro, Sindh, Pakistan

³ School of Mechatronics, Beijing Institute of Technology, Beijing 100081, China

Correspondence should be addressed to Arbab Nighat Khizer; arbabnighat@gmail.com

Received 31 March 2014; Revised 1 June 2014; Accepted 14 June 2014; Published 31 August 2014

Academic Editor: Qingsong Xu

Copyright © 2014 Arbab Nighat Khizer et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Stable hover flight control for small unmanned helicopter under light air turbulent environment is presented. Intelligent fuzzy logic is chosen because it is a nonlinear control technique based on expert knowledge and is capable of handling sensor created noise and contradictory inputs commonly encountered in flight control. The fuzzy nonlinear control utilizes these distinct qualities for attitude, height, and position control. These multiple controls are developed using two-loop control structure by first designing an inner-loop controller for attitude angles and height and then by establishing outer-loop controller for helicopter position. The nonlinear small unmanned helicopter model used comes from X-Plane simulator. A simulation platform consisting of MATLAB/Simulink and X-Plane[®] flight simulator was introduced to implement the proposed controls. The main objective of this research is to design computationally intelligent control laws for hovering and to test and analyze this autopilot for small unmanned helicopter model on X-Plane under ideal and mild turbulent condition. Proposed fuzzy flight controls are validated using an X-Plane helicopter model before being embedded on actual helicopter. To show the effectiveness of the proposed fuzzy control method and its ability to cope with the external uncertainties, results are compared with a classical PD controller. Simulated results show that two-loop fuzzy controllers have a good ability to establish stable hovering for a class of unmanned rotorcraft in the presence of light turbulent environment.

1. Introduction

Unmanned aerial vehicles (UAVs), particularly small unmanned helicopters, are gaining more interest from worldwide researchers due to their hovering ability and vertical takeoff and landing (VTOL) capability [1]. Demand for these capabilities has increased, especially in military and civil applications, which, in turn, results in greater demands on automated control systems [2]. Miniature helicopters come with various advantages such as small size, light weight, and low cost. However, they are under actuated, strongly coupled, and they have unmodeled dynamics and highly nonlinear systems [3, 4]. All these characteristics create significant challenges for flight control design including trajectory control and attitude control. Another major problem rarely addressed by researchers is that of the

presence of wind. Wind turbulence can create large thrust variations to the helicopter, which create a challenge to maintaining a constant height. Generally, an unmanned helicopter is controlled by a remote pilot or it is autonomous according to the preprogrammed flight plans created by sophisticated dynamic automation systems. A large number of effective flight control methods have been reported for small unmanned helicopter including robust adaptive control [5–7], H_∞ control [8–10], state-dependent Riccati equation control [11], sliding mode control (SMC) [12], trajectory tracking control [13], back stepping control [14, 15], fuzzy logic control [16, 17], and neural network control [18, 19]. Among these methods, fuzzy logic control has received significant attention from researchers because of its ability to handle nonlinearities [6, 20]. Recently, fuzzy control has been successfully proposed for control and navigation of

miniature VTOLs [21] and fixed wing UAVs [22]. Hybrid intelligent control systems were used for minihelicopter [23], addressing the stabilization of the helicopter without requiring Takagi-Sugeno methodology. Fuzzy control was used for landing systems along with autonomous flight using MATLAB and Aerosim aeronautical simulation block set [24]. A work on design of a fuzzy gain scheduled attitude/altitude controller for the unmanned helicopter under wind and gusts environment was reported in [16]. Flight testing of UAVS in real environment has always been expensive due to high aviation fuel costs. Additionally, new emerging technologies require extensive flight testing; therefore, real environment testing is too expensive and uneconomical. A viable alternative is to test the flight systems in a simulation-based environment. A commercially available flight simulator “X-Plane” can be used as an engineering tool to test and validate flight characteristics of fixed and rotary wing UAVs. Due to its built-in capability of communicating flight parameters, it has been widely used for modeling and control of UAVs [25–27].

In this paper, Mamdani-based fuzzy control is proposed for small unmanned helicopter at hovering under turbulent environment, which is the most important, complicated, and sensitive flight mode. Hovering refers to the ability to hold position, height, and heading during constant flight mode; therefore, helicopter control systems require guaranteed stability at first to hold the helicopter. In order to control the hover state, the control system is synthesized by applying pilot expertise in its simplest form by means of two-loop (inner-loop and outer-loop) fuzzy control. Inner-loop design contains pitch, roll, and yaw angles and height control while outer-loop constitutes the position control. In addition, external disturbance such as air turbulence is taken into account within the system model. The small Raptor-30 V2 helicopter is selected to test this proposed fuzzy control scheme comes from X-Plane simulator. It is the most realistic flight simulator in the flight aviation domain. Additionally, another main purpose of this research is to demonstrate the possibility of modifying and testing this intelligent flight control in a fast and easy way using an appropriate flight simulator tool such as X-Plane.

The paper is organized as follows. Section 2 describes the dynamical model of small unmanned helicopter. Section 3 discusses the simulation platform consisting of flight simulator X-Plane and MATLAB. Section 4 details proposed fuzzy laws for inner- and outer-loop control. Section 5 presents the simulation results. Finally, Section 6 outlined conclusion and future research work.

2. Dynamical Model of Small Unmanned Helicopter

The nonlinear model used for small helicopter is a physical model based on [28]. Newton's second law and Euler equations of motion of rotation are applied to helicopter model because it is considered as a rigid body. It is also important to determine force and torque vectors which are acting upon it. Therefore, the whole motion profile of a helicopter model

FIGURE 1: Helicopter model.

can be derived from the six degrees of motions equations. Three of them are translational motions and other three are assigned to rotational movement. The helicopter model shown in Figure 1 includes specific forces, velocities, angles, and related reference frames.

This paper assumed some suppositions in order to simplify and derive the final form of equations of motion. These are given as follows.

- (i) The inertial cross product is negligible.
- (ii) The effect of the main rotor induced air flow on the fuselage is negligible.
- (iii) There is always uniform and steady inflow at the main rotor.
- (iv) Air density is assumed to be constant in hovering mode.
- (v) Rotor blade distortions are neglected during the computation of rotor thrust.
- (vi) No twist is considered on the main rotor blades.
- (vii) Planes of symmetry exist along the zx -axis, therefore $I_{xy} = I_{yz} = 0$.
- (viii) Planes of symmetry exist for mass along the YZ -axis, this implies $I_{xz} = 0$.

Small helicopter mathematical model is described in [28] and is divided into different submodels which interact with each other because of heavy dynamical coupling. These small submodels can be divided into two categories; one is helicopter controls and other refers to helicopter body dynamics as shown in Figure 2.

2.1. Helicopter Control Surfaces. A helicopter has four control inputs to control attitude and movement. The first aileron servo control input (δ_{lat}) controls the lateral cyclic pitch on the main rotor which results in roll motion that makes the helicopter moves in the lateral direction. The second elevator servo input (δ_{long}) controls the longitudinal cyclic pitch on the main rotor which results in pitch motion that makes the helicopter moves in the longitudinal direction. The third collective pitch (δ_{coll}) controls the collective pitch on the

FIGURE 2: Integrated helicopter model.

FIGURE 3: Helicopter control surfaces.

main rotor which results in heave motion that makes the helicopter moves in the vertical direction. Finally the rudder servo input (δ_{ped}) controls the collective pitch on the tail rotor which results in yaw motion as shown in Figure 3.

2.2. Main Rotor Model. The primary source of forces and moments generation on the helicopter is due to rotation of main rotor and which usually consists of two blades and one hub. When it rotates, the blade and air-relative motion produces air power, while the hub is used to connect the blade and rotational axis. Based on classical momentum theory, thrust, induced velocity can be given as follows [29]:

$$T_{\text{mr}} = (w_{\text{blade}} - v_i) \cdot \frac{\rho \Omega R^2 a_{\text{mr}} Bc}{4},$$

$$v_i^2 = \sqrt{\left(\frac{\hat{v}^2}{2}\right)^2 + \left(\frac{T_{\text{mr}}}{2 \cdot \rho \cdot A}\right)^2} - \frac{\hat{v}^2}{2},$$

$$w_{\text{blade}} = w_r + \frac{2}{3} \Omega \cdot R \left[\delta_{\text{coll}} + \frac{3}{4} \theta_{\text{twist}} \right],$$

$$w_r = w + (a_{1s} + i_s) \cdot u - b_{1s} \cdot v,$$

$$\hat{v}^2 = u^2 + v^2 + w^2 \cdot (w_r - 2v_i), \quad (1)$$

$$A = \pi R^2.$$

The description of all variables is given in nomenclature. Moment equations are given as

$$L = (K_s + T_{\text{mr}} h_{\text{mr}}) \sin(b),$$

$$M = (K_s + T_{\text{mr}} h_{\text{mr}}) \sin(a),$$

$$N = -C_q A \Omega^2 R^3 \rho,$$

$$C_q = \frac{C_{\text{mr}} (v_i - w)}{\Omega R} + C_{d0} bc \left[\frac{1 + 7(u^2 + v^2)/3\Omega^2 R^2}{8\pi R} \right]. \quad (2)$$

The flapping dynamic equations of the main rotor are expressed in [30] as

$$\dot{a} = -q - \frac{a}{\tau_f} + \frac{A_{\text{lon}}}{\tau_f} \delta_{\text{lon}},$$

$$\dot{b} = -p - \frac{b}{\tau_f} + \frac{B_{\text{lat}}}{\tau_f} \delta_{\text{lat}}. \quad (3)$$

2.3. Tail Rotor Model. The tail rotor is used to compensate the torque about z -axis by producing a counter side force and to help in course correction. The helicopter is equipped with a gyroscope which controls tail rotor blade pitch needed to maintain angular velocity (r) at zero as well as adjust the thrust of tail rotor so that the helicopter does not rotate about the z -axis. As a result, the helicopter is only able to yaw by

FIGURE 4: Tail rotor system.

using the pedal input controlling the tail rotor. The tail rotor model is given in Figure 4. Consider

$$\begin{aligned}
T_{\text{tr}} &= (w_{\text{btr}} - v_{\text{itr}}) \cdot \frac{\rho \Omega R^2 a_{\text{tr}} B c}{4}, \\
v_{\text{itr}}^2 &= \sqrt{\left(\frac{\hat{v}_{\text{tr}}^2}{2}\right)^2 + \left(\frac{T_{\text{tr}}}{2 \cdot \rho \cdot A}\right)^2} - \frac{\hat{v}_{\text{tr}}^2}{2}, \\
w_{\text{btr}} &= w_{\text{tr}} + \frac{2}{3} \Omega \cdot R \left[\delta_{\text{ped}} + \frac{3}{4} \theta_{\text{twist}} \right], \\
w_{\text{tr}} &= v - r d_{\text{tr}} + p h_{\text{tr}}, \\
\hat{v}_{\text{tr}}^2 &= u^2 + (w + q d_{\text{tr}})^2 + w_{\text{tr}} (w_{\text{tr}} - 2 v_{\text{itr}}), \\
Y_{\text{tr}} &= -T_{\text{tr}}, \\
L_{\text{tr}} &= Y_{\text{tr}} h_{\text{tr}}, \\
N_{\text{tr}} &= -Y_{\text{tr}} d_{\text{tr}}.
\end{aligned}
\tag{4}$$

2.4. Fuselage Model. The fuselage model equations are defined under the condition $v_{\text{trim}} = \sqrt{u_a^2 + v_a^2 + w_a^2}$:

$$\begin{aligned}
\mathbf{v}_{\text{trim}} &< \mathbf{v}_i \\
X_{\text{fus}} &= -0.5\rho S_{\text{fus}_x} u_a v_i, \\
Y_{\text{fus}} &= -0.5\rho S_{\text{fus}_y} v_a v_i, \\
Z_{\text{fus}} &= -0.5\rho S_{\text{fus}_z} (w_a - v_i) v_i, \\
\mathbf{v}_{\text{trim}} &\geq \mathbf{v}_i \\
X_{\text{fus}} &= -0.5\rho S_{\text{fus}_x} u_a v_{\text{trim}}, \\
Y_{\text{fus}} &= -0.5\rho S_{\text{fus}_y} v_a v_{\text{trim}}, \\
Z_{\text{fus}} &= -0.5\rho S_{\text{fus}_z} (w_a - v_i) v_{\text{trim}}.
\end{aligned}
\tag{5}$$

2.5. Vertical Fin Model. The vertical fin model describes the force which is produced by tail rotor around the z -axis of the

system and affects the behavior of helicopter body according to [31]. Consider

$$\begin{aligned} Y_{vf} &= 0.5 \rho S_{vf} \left(C_{la}^{vf} |u_a| v_a + |u_a| u_a \right), \\ |Y_{vf}| &= 0.5 \rho S_{ht} \left(u_a^2 + v_a^2 \right), \\ N_{vf} &= -Y_{vf} d_{vf}. \end{aligned} \quad (6)$$

2.6. Rigid Body Dynamics and Attitude Kinematics. Consider

$$\dot{u} = (vr - wq) - g \sin \theta + \frac{F_x}{m} \quad (7)$$

$$\dot{v} = (wp - ur) + g \cos \theta \sin \phi + \frac{F_y}{m} \quad (8)$$

$$\dot{w} = (uq - vp) + g \cos \theta \cos \phi + \frac{F_z}{m} \quad (9)$$

$$\dot{p} = \frac{qr(I_{yy} - I_{zz}) + L}{I_{xx}} \quad (10)$$

$$\dot{q} = \frac{rp(I_{zz} - I_{xx}) + M}{I_{yy}} \quad (11)$$

$$\dot{r} = \frac{pq(I_{xx} - I_{yy}) + N}{I_{zz}} \quad (12)$$

$$\dot{\phi} = p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \quad (13)$$

$$\dot{\theta} = q \cos \phi - r \sin \phi \quad (14)$$

$$\psi = q \sin \phi \sec \theta + r \cos \phi \sec \theta. \quad (15)$$

3. Simulation Platform

The simulation platform is implemented using MATLAB and X-Plane, a commercially available flight simulator developed by Laminar Research. In this work X-Plane is used to provide the model of small unmanned helicopter control and to simulate its dynamics. Two computers are used for this simulation platform and communication is done through a UDP (user datagram protocol) data communication bus as shown in Figure 5.

3.1. X-Plane Flight Simulator. X-Plane flight simulator was chosen because it is certified by the Federal Aviation Administration (FAA) to train pilots [32] and its aircraft models are extremely accurate which ensures the simulation with high degree of accuracy as an actual flight. Apart from its realistic flight simulation capability, X-Plane also incorporates additional useful features which make it an ideal choice for experimenting and validating UAVs. It has plane marker tool which uses engineering method known as blade element theory, allows the division of aircraft model into small elements and then finds the forces acting on each element several times per second [33]. The control algorithm is developed and stored in Simulink. During each sampling

FIGURE 5: Simulation platform.

FIGURE 6: X-Plane input/output data selecting window.

period, the control algorithm receives signals from X-Plane and outputs the control commands to X-Plane through the UDP connection. Figure 6 depicts the X-Plane input/output data window used to perform a simulation and allows the user to adjust the update rate from 1 to 99 packets per second in UDP rate.

The communication speed between software is 40 packets/second were selected for this work. The UDP packet size and type depend on the amount of data being exported. Generally, the pattern of data packets being sent by X-Plane is of the following format: the first four bytes indicate the type of packet i-e “DATA,” the fifth byte is used for an internal policy, the next incoming four bytes represent what type of parameter is being sent to Simulink, and the following four bytes represent the value in single-precision float [34]. A pattern of data packet is shown in Figure 7.

3.2. MATLAB/Simulink. MATLAB has user graphical environment and a libraries set that allow users to design, simulate, implement, and test a variety of time-varying systems relating to controls, communications, and signal/video/image processing areas. Simulink is integrated with MATLAB, which provides instant access to an extensive range of tools used to develop algorithms, analyze and visualize simulations, customize the modeling environment, and define the signal and parameters of data tested [35].

In this simulation platform, the MATLAB /Simulink is used as software which is responsible for processing the UDP data to and from X-Plane and controls the system for

processing of data simulation. Figure 8 shows how MATLAB /Simulink and X-Plane are interconnected for this hovering control system.

4. Fuzzy Control Design

The proposed fuzzy system for hovering mode under light air turbulent environment is composed of two-loops, i-e inner-loop and outer-loop, responsible for attitude, height, and position control, respectively. First, attitude angles are used as feedback information in order to keep the helicopter stable. Second, after achieving stability through attitude angles, a height fuzzy controller is designed to control the altitude of the helicopter with high holding ability. Finally, based upon positional feedback information, an outer-loop position fuzzy controller is designed. The fuzzy inference system (FIS) for hovering mode is shown in (16) and an overall fuzzy control structure is depicted in Figure 9. Consider

$$\begin{aligned}
 C_{\text{coll}} &= \text{FIS}(\text{Height}), \\
 C_{\text{long}} &= \text{FIS}(\text{pitch}), \\
 C_{\text{lat}} &= \text{FIS}(\text{roll}), \\
 C_{\text{ped}} &= \text{FIS}(\text{yaw}), \\
 C_{\text{pitch}} &= \text{FIS}(\text{longitudinal}), \\
 C_{\text{roll}} &= \text{FIS}(\text{lateral}).
 \end{aligned} \tag{16}$$

Assume that the helicopter is working clockwise and its operation phenomena are based on the experiences of a human pilot [36, 37]. Consider the following.

- (i) While the helicopter pitching angle is negative, increase the longitudinal cycle variable which is responsible for making the pitch angle positive so that balanced attitude can be achieved.
- (ii) While the helicopter fuselage is deflecting towards the left, increase latitudinal cycle variable pitch positively to balance the attitude.
- (iii) While increasing the collective positively, the pedal will also be increased positively to keep the yaw balanced.
- (iv) While the helicopter vertical position and velocity are negative, increase collective variable to hold the helicopter at desired height.

4.1. Inner-Loop Control. Inner-loop control constitutes of attitude and height Mamdani fuzzy control for small helicopter flight and is further divided into three angles controller i-e pitch, roll, and yaw control. Attitude angular velocity and z -position are considered to be input while cyclic pitch (longitudinal and lateral both), tail rotor pitch, and collective pitch are an output of the inner-loop control. The inner-loop control structure is depicted in Figure 10.

	Data				I	Label				Group-1				Group-2				Group-2 ... Group-8			
Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	—	—	41
						Index number 8				Elevator				Aileron				Rudder ...			

FIGURE 7: UDP packet format.

FIGURE 8: Simulink-X-Plane interconnection.

FIGURE 9: Overall fuzzy control structure.

FIGURE 10: Inner-loop control structure.

FIGURE 11: Pitch angle fuzzy control framework.

FIGURE 12: Input and output Gaussian MF of pitch fuzzy control.

FIGURE 13: Input and output Gaussian MF of height fuzzy control.

4.1.1. Pitch Fuzzy Control. The pitch fuzzy controller is mainly concerned with the longitudinal cyclic pitch as an output and pitch angle error (e_θ) and pitch rate (q) as input is shown in Figure 11 where g_1 and g_2 are input quantifying factors and g_o is the output scaling factor.

The inputs and output of pitch control are divided into seven, five, and nine fuzzy intervals, respectively, as shown in Figure 12 to ensure the control sensitivity and stability.

FIGURE 14: Signs definition of variables used in lateral and longitudinal motion.

Gaussian MF is selected due to its higher smoothness level than triangular MF for input and output variables. The associated rules are given in the appendix.

The other two roll and yaw fuzzy controls are designed in a same way and their detailed rule description is available in the appendix.

4.1.2. Height Fuzzy Control. For height fuzzy control, it is assumed that main rotor speed has constant value; therefore the collective control corresponds to the error of vertical position (z) and vertical rate (w). The height input and output MF are shown in Figure 13 and rules are given in the appendix.

4.2. Outer-Loop Control. The outer-loop includes longitudinal and lateral position control of small unmanned helicopter. The control is mainly handled by main rotor longitudinal cyclic pitch and lateral cyclic pitch movement. The horizontal and vertical position fuzzy controller generates a desired attitude angle (θ_d, ϕ_d), which is further used as input to attitude fuzzy controller, in order to achieve stability. Longitudinal positional error (e_x, e_y) and velocity (u, v) is the input for the controller and desired pitch and roll angle is the output. Again Gaussian MF is selected for both input and output and rules are formulated on a heuristic basis. Finally, two fuzzy controllers are designed according to signs definition depicted in Figure 14, which effectively determines the desired values for the pitch and roll angle. Figure 15 shows the proposed outer-loop control architecture.

4.2.1. Longitudinal Position Control. Given x -position as a reference, this longitudinal control infers the desired pitch

FIGURE 15: Outer-loop position control structure.

FIGURE 16: Input and output Gaussian MF of longitudinal fuzzy control.

angle (θ_d) using the error of x -position and translational velocity (u). The input and output MF are shown in Figure 16 and twenty-five analogous fuzzy rules used to determine the desired pitch angle are given in the appendix. The interpretation of first rule is if the error in the position is negative and its rate of change is also negative then it means that a positive pitch angle is causing the helicopter to move away from the desired position therefore a relatively large negative pitch angle is required to slow down the longitudinal motion, reverse the velocity, and reduce the error in position. All rules are explained in this way.

4.2.2. Lateral Position Control. Given y -position as a reference, the lateral control infers the desired roll angle (ϕ_d) using the error of y -position and translational velocity (v). The input, output MF and analogous fuzzy rules are used to determine the desired roll angle and are given in the appendix. The first rule can be interpreted as follows: if the error of y -position is negative and its derivative is also negative it means that negative roll angle is causing the helicopter to move away from the desired position therefore a relatively large positive roll angle is required to slow down

the lateral motion and as a result the velocity is reversed and causes a reduction in the position error. All other rules are interpreted in a similar fashion.

5. Simulation Results

The unmanned helicopter model used in the simulation platform is the Raptor-30 V2 RC helicopter. The Raptor-30 V2 is a highly maneuverable helicopter weighing 3 kg. The fuselage length is 1,150 mm and its width is approximately 140 mm.

MATLAB and X-Plane are operated on two computers and the IP addresses and port numbers used for sending and receiving packets are 10.108.29.17 and 10.108.28.65 and 49000 and 49001, respectively. The overall flight control system implemented in Simulink, using the UDP interface that communicates with X-Plane, is shown in Figures 17(a) and 17(b). In the simulation, first the system was test under the ideal weather condition without wind and randomness and then a light turbulent weather of 10 knots (4.5 m/s) wind speed, 2 knots (0.9 m/s) shear speed, and level 3 turbulence was introduced. The details of the ideal and light turbulent weather conditions are shown in Figures 18(a) and 18(b).

The mean and standard deviations of the positions are $(1.57 + 04, 749.63, -1.811 + 04)$ and $(48.49, 55.02, 215.6630)$ cm, respectively, for ideal weather. The corresponding mean and standard deviations of hovering heading are $(2.49, -0.28, 161.1)$ and $(1.2, 1.26, 5.26)$ deg. The fuzzy control is able to control the Raptor-30 V2 unmanned helicopter to hover on this specific height as shown in Figure 19.

The mean and standard deviations of the positions are $(1.52 + 04, 104.18, -1.831 + 04)$ and $(42.49, 37.35, 217.6630)$ cm, respectively, for light turbulent weather. Fuzzy control is able to control the Raptor-30 V2 unmanned helicopter to hover and it is robust to operate in the real environment as shown in Figure 20.

When helicopter model hovers, the main rotors rotate and generate a moment which causes the rotation of body around the z -axis. This moment is counteracted through the tail rotor by producing a force in the y -axis (negative direction). This action accelerates the helicopter in the same direction, thus increasing the velocity in the same direction and causes the blades to flap positive lateral. When the

(a)

(b)

FIGURE 17: Overall system architecture in Simulink.

FIGURE 18: (a) Ideal weather settings in X-Plane; (b) light turbulent weather settings in X-Plane.

FIGURE 19: Hovering test result of the fuzzy control under the ideal weather condition.

FIGURE 20: Hovering test result of the fuzzy control under light turbulent weather condition.

lateral flapping becomes positive, this results in longitudinal flapping which becomes negative because of the cross-coupling. The lateral flapping also makes the helicopter rotate about the x -axis (p becomes positive) and causes the velocity along y -axis to become positive (v becomes positive). The negative longitudinal flapping causes the helicopter body to rotate negatively about y -axis (q becomes negative) and thereby giving a positive translational velocity along x -axis (u becomes positive). Figure 21 depicts this situation and Figure 22 shows the surface inputs to the helicopter model. As mentioned above, this fuzzy control scheme is tested using the X-Plane flight simulator model, which requires six degrees for its hover at initial levels.

Figure 23 portrays the fuzzy control response for attitude angles used in tracking the reference trajectories. Pitch and roll angles are oscillating in order to maintain these tracking signals. The oscillation of the yaw angle is due to the coupling between two inputs (collective and pedal). Figure 24 depicts

the performance of proposed fuzzy two-loop controller with conventional PD control under light turbulent weather condition. The conventional PD performance observed significantly inferior compared to the fuzzy control. The flight results indicate that the fuzzy controller exhibits satisfactory behavior in the presence of light air turbulence.

Figure 25 shows a visual presentation of the X-Plane Rapter-30 V2 hovering using simple and robust fuzzy control.

Comment 1. It should be highlighted that the originality of this research work is the design of simple and efficient control systems for a small unmanned helicopter model Rapter-30 V2 using X-Plane flight simulator at hover. The simulation platform implements two computers with an autopilot system applied by the X-Plane model which is similar to a real rotorcraft, thus minimizing the damage risks and increasing the flexibility for on the spot design changes. Simplicity is achieved through SISO Mamdani controller

FIGURE 21: Translational and angular velocities.

FIGURE 22: Surface Inputs.

FIGURE 23: Attitude fuzzy control response for tracking signals.

FIGURE 24: Hovering maneuver: reference position trajectory (black solid line) and actual position trajectory of the fuzzy (red solid line), PD (blue dashed line) controller with respect to the inertial axis.

FIGURE 25: Rapter-30 V2 at hovering.

FIGURE 26: Roll input and output MF.

FIGURE 27: Yaw input and output MF.

FIGURE 28: Lateral position input and output MF.

design. The Mamdani fuzzy approach is selected here instead of Takagi-Sugeno approach, because it permits the inclusion of experimental knowledge for the controller synthesis, which is available from the flight of the miniature helicopter by a human pilot.

6. Conclusion

This paper attempts to introduce a dynamical model of a small scale unmanned helicopter in detail. Using an expert knowledge and pilot-based experience, a stable inner-loop and outer-loop fuzzy control for attitude, height, and position were designed to control the direction, height, and translational motion of the small unmanned helicopter. The flight control system is implemented in MATLAB/Simulink in one computer and X-Plane helicopter model on second computer to investigate the performance of fuzzy control at hovering. The communication between two computers is done by UDP using Ethernet connection. To show the effectiveness of proposed two-loop controller, light turbulent weather was introduced. Simulation results clearly shown that proposed fuzzy control successfully achieved hovering control under ideal and mild turbulent weather. The satisfactory performance of fuzzy control compared to conventional PD design was attributed to the attenuated cross-coupling effects amongst the Raptor-30 V2 dynamics. The proposed controller was able to control the small Raptor-30 V2 at desired height while keeping stable direction. The simulation results shows that the unmanned helicopter tracks an imposed trajectory with good overall performance. The implementation of intense weather condition is the next objective of work.

Appendix

NB (negative big), NM (negative medium), NS (negative small), ZE (zero), PS (positive small), PM (positive medium), PB (positive big), ZP (zero positive), and ZN (zero negative).

Pitch.Fis Rule.

- (1) If (epitch is NB) and (pitch_rate is NB) then (cyclic (long) is PB) (1)
- (2) If (epitch is NB) and (pitch_rate is NS) then (cyclic (long) is PM) (1)
- (3) If (epitch is NB) and (pitch_rate is ZE) then (cyclic (long) is PM)(1)
- (4) If (epitch is NB) and (pitch_rate is PS) then (cyclic (long) is PS) (1)
- (5) If (epitch is NB) and (pitch_rate is PB) then (cyclic (long) is ZP) (1)
- (6) If (epitch is NM) and (pitch_rate is NB) then (cyclic (long) is PB) (1)
- (7) If (epitch is NM) and (pitch_rate is NS) then (cyclic (long) is PM) (1)
- (8) If (epitch is NM) and (pitch_rate is ZE) then (cyclic (long) is PS) (1)

- (9) If (epitch is NM) and (pitch_rate is PS) then (cyclic (long) is ZP) (1)
- (10) If (epitch is NM) and (pitch_rate is PB) then (cyclic (long) is ZE) (1)
- (11) If (epitch is NS) and (pitch_rate is NB) then (cyclic (long) is PM) (1)
- (12) If (epitch is NS) and (pitch_rate is NS) then (cyclic (long) is PS) (1)
- (13) If (epitch is NS) and (pitch_rate is ZE) then (cyclic (long) is ZP) (1)
- (14) If (epitch is NS) and (pitch_rate is PS) then (cyclic (long) is ZE) (1)
- (15) If (epitch is NS) and (pitch_rate is PB) then (cyclic (long) is NM) (1)

⋮

- (29) If (epitch is PM) and (pitch_rate is PS) then (cyclic (long) is NM) (1)
- (30) If (epitch is PM) and (pitch_rate is PB) then (cyclic (long) is NB) (1)
- (31) If (epitch is PB) and (pitch_rate is NB) then (cyclic (long) is ZN) (1)
- (32) If (epitch is PB) and (pitch_rate is NS) then (cyclic (long) is NS) (1)
- (33) If (epitch is PB) and (pitch_rate is ZE) then (cyclic (long) is NM) (1)
- (34) If (epitch is PB) and (pitch_rate is PS) then (cyclic (long) is NM) (1)
- (35) If (epitch is PB) and (pitch_rate is NB) then (cyclic (long) is NB) (1)

(i) Roll.FIS

See Figure 26.

Roll.FIS Rule Base.

- (1) If (error_roll_angle is NB) and (roll_rate is NB) then (cyclic (lat) is PB) (1)
 - (2) If (error_roll_angle is NB) and (roll_rate is NS) then (cyclic (lat) is PM) (1)
 - (3) If (error_roll_angle is NB) and (roll_rate is ZE) then (cyclic (lat) is PM) (1)
 - (4) If (error_roll_angle is NB) and (roll_rate is PS) then (cyclic (lat) is PS) (1)
 - (5) If (error_roll_angle is NB) and (roll_rate is PB) then (cyclic (lat) is ZP) (1)
- ⋮
- (29) If (error_roll_angle is PM) and (roll_rate is PB) then (cyclic (lat) is NB) (1)

- (30) If (error_roll_angle is PB) and (roll_rate is NB) then (cyclic (lat) is ZN) (1)
- (31) If (error_roll_angle is PB) and (roll_rate is NS) then (cyclic (lat) is NS) (1)
- (32) If (error_roll_angle is PB) and (roll_rate is ZE) then (cyclic (lat) is NM) (1)
- (33) If (error_roll_angle is PB) and (roll_rate is PS) then (cyclic (lat) is NM) (1)
- (34) If (error_roll_angle is PB) and (roll_rate is PB) then (cyclic (lat) is NB) (1)
- (35) If (error_roll_angle is ZE) and (roll_rate is ZE) then (cyclic (lat) is ZE) (1)

(ii) Yaw.FIS

See Figure 27.

Yaw.FIS Rule Base.

- (1) If (error_yaw_angle is NB) and (yawrate is NB) then (pedal is NB) (1)
 - (2) If (error_yaw_angle is NB) and (yawrate is NM) then (pedal is NM) (1)
 - (3) If (error_yaw_angle is NB) and (yawrate is NS) then (pedal is NM) (1)
 - (4) If (error_yaw_angle is NB) and (yawrate is ZE) then (pedal is NS) (1)
 - (5) If (error_yaw_angle is NB) and (yawrate is PS) then (pedal is NS) (1)
 - (6) If (error_yaw_angle is NB) and (yawrate is PM) then (pedal is ZN) (1)
 - (7) If (error_yaw_angle is NB) and (yawrate is PB) then (pedal is ZE) (1)
 - (8) If (error_yaw_angle is NM) and (yawrate is NB) then (pedal is NM) (1)
- ⋮
- (40) If (error_yaw_angle is PM) and (yawrate is PS) then (pedal is PS) (1)
 - (41) If (error_yaw_angle is PM) and (yawrate is PM) then (pedal is PS) (1)
 - (42) If (error_yaw_angle is PM) and (yawrate is PB) then (pedal is PM) (1)
 - (43) If (error_yaw_angle is PB) and (yawrate is NB) then (pedal is ZE) (1)
 - (44) If (error_yaw_angle is PB) and (yawrate is NM) then (pedal is ZP) (1)
 - (45) If (error_yaw_angle is PB) and (yawrate is NS) then (pedal is PS) (1)
 - (46) If (error_yaw_angle is PB) and (yawrate is ZE) then (pedal is PS) (1)
 - (47) If (error_yaw_angle is PB) and (yawrate is PS) then (pedal is PM) (1)

(48) If (error_yaw_angle is PB) and (yawrate is PM) then (pedal is PM) (1)

(49) If (error_yaw_angle is PB) and (yawrate is PB) then (pedal is PB) (1)

(iii) Longitudinal position. FIS Rule Base

(1) If (ex is NB) and (u is NB) then (long_{u} is NB) (1)

(2) If (ex is NB) and (u is NS) then (long_{u} is NM) (1)

(3) If (ex is NB) and (u is ZE) then (long_{u} is NM) (1)

(4) If (ex is NB) and (u is PS) then (long_{u} is NS) (1)

(5) If (ex is NB) and (u is PB) then (long_{u} is ZN) (1)

(6) If (ex is NM) and (u is NB) then (long_{u} is NB) (1)

(7) If (ex is NM) and (u is NS) then (long_{u} is NM) (1)

(8) If (ex is NM) and (u is ZE) then (long_{u} is NS) (1)

(9) If (ex is NM) and (u is PS) then (long_{u} is NS) (1)

(10) If (ex is NM) and (u is PB) then (long_{u} is ZN) (1)

⋮

(21) If (ex is PS) and (u is NB) then (long_{u} is ZP) (1)

(22) If (ex is PS) and (u is NS) then (long_{u} is ZP) (1)

(23) If (ex is PS) and (u is ZE) then (long_{u} is PS) (1)

(24) If (ex is PS) and (u is PS) then (long_{u} is PM) (1)

(25) If (ex is PS) and (u is PB) then (long_{u} is PB) (1)

(26) If (ex is PM) and (u is NB) then (long_{u} is ZP) (1)

(27) If (ex is PM) and (u is NS) then (long_{u} is PS) (1)

(28) If (ex is PM) and (u is ZE) then (long_{u} is PS) (1)

(29) If (ex is PM) and (u is PS) then (long_{u} is PM) (1)

(30) If (ex is PM) and (u is PB) then (long_{u} is PB) (1)

(31) If (ex is PB) and (u is NB) then (long_{u} is ZP) (1)

(32) If (ex is PB) and (u is NS) then (long_{u} is PS) (1)

(33) If (ex is PB) and (u is ZE) then (long_{u} is PM) (1)

(34) If (ex is PB) and (u is PS) then (long_{u} is PM) (1)

(35) If (ex is PB) and (u is PB) then (long_{u} is PB) (1)

(iv) Lateral position. FIS Rule Base

See Figure 28.

Lateral Position. FIS Rule Base.

(1) If (ey is NB) and (v is NB) then (lat_{v} is PB) (1)

(2) If (ey is NB) and (v is NS) then (lat_{v} is PM) (1)

(3) If (ey is NB) and (v is ZE) then (lat_{v} is PS) (1)

(4) If (ey is NB) and (v is PS) then (lat_{v} is ZP) (1)

(5) If (ey is NB) and (v is PB) then (lat_{v} is ZE) (1)

(6) If (ey is NS) and (v is NB) then (lat_{v} is PM) (1)

(7) If (ey is NS) and (v is NS) then (lat_{v} is PS) (1)

(8) If (ey is NS) and (v is ZE) then (lat_{v} is ZP) (1)

⋮

(20) If (ey is PS) and (v is PB) then (lat_{v} is NM) (1)

(21) If (ey is PB) and (v is NB) then (lat_{v} is ZE) (1)

(22) If (ey is PB) and (v is NS) then (lat_{v} is ZN) (1)

(23) If (ey is PB) and (v is ZE) then (lat_{v} is NS) (1)

(24) If (ey is PB) and (v is PS) then (lat_{v} is NM) (1)

(25) If (ey is PB) and (v is PB) then (lat_{v} is NB) (1)

(v) Height. FIS Rule Base

(1) If (ez is NB) and (w is NB) then (collective is NB) (1)

(2) If (ez is NB) and (w is NM) then (collective is NM) (1)

(3) If (ez is NB) and (w is NS) then (collective is NM) (1)

(4) If (ez is NB) and (w is ZE) then (collective is NS) (1)

(5) If (ez is NB) and (w is PS) then (collective is NS) (1)

(6) If (ez is NB) and (w is PM) then (collective is ZN) (1)

(7) If (ez is NB) and (w is PB) then (collective is ZE) (1)

(8) If (ez is NM) and (w is NB) then (collective is NB) (1)

(9) If (ez is NM) and (w is NM) then (collective is NM) (1)

(10) If (ez is NM) and (w is NS) then (collective is NS) (1)

⋮

⋮

(36) If (ez is PM) and (w is NB) then (collective is ZN)
(1)

(37) If (ez is PM) and (w is NM) then (collective is ZE)
(1)

(38) If (ez is PM) and (w is NS) then (collective is ZP)
(1)

(39) If (ez is PM) and (w is ZE) then (collective is PS)
(1)

(40) If (ez is PM) and (w is PS) then (collective is PS)
(1)

(41) If (ez is PM) and (w is PM) then (collective is PM)
(1)

(42) If (ez is PM) and (w is PB) then (collective is PM)
(1)

(43) If (ez is PB) and (w is NB) then (collective is ZE)
(1)

(44) If (ez is PB) and (w is NM) then (collective is ZP)
(1)

(45) If (ez is PB) and (w is NS) then (collective is PS)
(1)

Nomenclature

O_b, X_b, Y_b, Z_b :	Body reference frame (BRF)
O_g, X_g, Y_g, Z_g :	Ground reference frame (GRF)
$L_{(\cdot)^*}, M_{(\cdot)^*}, N_{(\cdot)^*}$:	Moments along the BRF
C_{d0} :	Main rotor blade drag coefficient
$\Omega_{(\cdot)^*}$:	Main/tail rotor rotating speed
τ_f :	Main rotor time constant
v_i :	Main rotor induced velocity
p, q, r :	Angular velocities
$T_{(\cdot)^*}$:	Main/tail rotor thrust
v_i :	Induced velocity
A :	Rotor disk area
β :	Blade flapping angle
(p, q, r) :	Angular helicopter body rates, in body frame
$\hat{v}$:	Total airspeed
w_{blade} :	Average velocity of main rotor blade relative to air
C_{d0} :	drag coefficient of main rotor
A_{lon}, B_{lat} :	Effective steady state lateral and longitudinal gains
R_{mr} :	Main rotor blade radius
F_x, F_y, F_z :	Total forces along x, y, and z-axes
I_{xx}, I_{yy}, I_{zz} :	Moments of inertia about x, y, and z-axes
τ_f :	Main rotor blade constant
θ, ϕ, ψ :	Euler angles

a, b :	Tip-path-plane flapping angle for longitudinal and lateral
K_s :	Main rotor spring constant
$h_{(\cdot)^*}$:	Distance between the rotor head and helicopter COG
ρ :	Air density
B :	Main rotor blade number
$c_{(\cdot)^*}$:	Main/tail blade chord length
δ_{coll} :	Collective pitch servo input
δ_{ped} :	Rudder servo input
δ_{lat} :	Aileron servo input
δ_{long} :	Elevator servo input
i_s :	Shaft tilt angle
(u, v, w) :	Velocity components relative to air defined in body-frame
θ_{twist} :	Blade twist
h_{tr} :	Tail rotor hub's vertical position above COF
C_q :	Intermediate variable
d_{tr} :	Tail rotor hub's longitudinal position behind COG
$a_{(\cdot)^*}$:	Blade lift curve slope
p, q, r :	Angular velocities
A_{lon}, B_{lat} :	Longitudinal and lateral steady state gains
w_r :	Rotor disk velocity relative to air
$(\cdot)^*$:	Related rotor (i-e main rotor or tail rotor).

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] P. Spanoudakis, N. Doitsidis, N. Tsourveloudis, and K. P. Valavanis, "A market overview of the vertical take-off and landing UAVs," in *Proceedings of the Workshop on Unmanned Aerial Vehicles 11th Mediterranean Conference on Control and Automation*, Rhodes, Greece, 2003.
- [2] K. P. Valvanis, Ed., *Advances in Unmanned Aerial Vehicles*, Springer, New York, NY, USA, 1st edition, 2007.
- [3] M. B. Bragg, T. Basar, W. R. Perkins et al., "Smart icing systems for aircraft icing safety," in *Proceedings of the 40th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA, Reno, Nev, USA, 2002.
- [4] P. Castillo, Ed., *Modelling and Control of Mini-Flying Machines*, Springer, New York, NY, USA, 1st edition, 2005.
- [5] E. N. Johnson and S. K. Kannan, "Adaptive flight control for an autonomous unmanned helicopter," in *AIAA Guidance, Navigation, and Control Conf. Exhibit*, August 2002.
- [6] A. Isidori, L. Marconi, and A. Serrani, "Robust nonlinear motion control of a helicopter," *IEEE Transactions on Automatic Control*, vol. 48, no. 3, pp. 413–426, 2003.
- [7] S. Lee, C. Ha, and B. S. Kim, "Adaptive nonlinear control system design for helicopter robust command augmentation," *Aerospace Science and Technology*, vol. 9, no. 3, pp. 241–251, 2005.
- [8] M. la Civita, G. Papageorgiou, W. C. Messner, and T. Kanade, "Design and flight testing of an H_∞ controller for a robotic

- helicopter," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 2, pp. 485–494, 2006.
- [9] C. C. LUO, R. F. LIU, C. D. YANG, and Y. H. CHANG, "Helicopter H_∞ control design with robust flying quality," *Aerospace Science and Technology*, vol. 7, no. 2, pp. 159–169, 2003.
 - [10] J. Gadewadikar, "Structured H_∞ command and control loop design for unmanned helicopters," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 4, pp. 1093–1102, 2008.
 - [11] A. Bogdanov and E. Wan, "State-dependent riccati equation control for small autonomous helicopters," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 47–60, 2007.
 - [12] H. Sira-Ramirez, M. Zribi, and S. Ahmad, "Dynamical sliding mode control approach for vertical flight regulation in helicopters," *IEEE Proceedings: Control Theory and Applications*, vol. 141, no. 1, pp. 19–24, 1994.
 - [13] J. Shan, H. T. Liu, and S. Nowotny, "Synchronised trajectory-tracking control of multiple 3-DOF experimental helicopters," *IEEE Proceedings: Control Theory and Applications*, vol. 152, no. 6, pp. 683–692, 2005.
 - [14] E. N. Johnson and S. K. Kannan, "Adaptive trajectory control for autonomous helicopters," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 3, pp. 524–538, 2005.
 - [15] S. S. Ge, B. Ren, K. P. Tee, and T. H. Lee, "Approximation-based control of uncertain helicopter dynamics," *IET Control Theory Applications*, vol. 3, no. 7, pp. 941–956, 2009.
 - [16] B. Kadmiry and D. Driankov, "A fuzzy gain-scheduler for the attitude control of an unmanned helicopter," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 4, pp. 502–515, 2004.
 - [17] B. Kadmiry and D. Driankov, "A fuzzy flight controller combining linguistic and model-based fuzzy control," *Fuzzy Sets and Systems*, vol. 146, no. 3, pp. 313–347, 2004.
 - [18] S. S. Ge, B. Ren, and K. P. Tee, "Adaptive neural network control of helicopters with unknown dynamics," in *Proceedings of the 45th IEEE Conference on Decision and Control (CDC '06)*, pp. 3022–3027, December 2006.
 - [19] R. Enns and J. Si, "Helicopter trimming and tracking control using direct neural dynamic programming," *IEEE Transactions on Neural Networks*, vol. 14, no. 4, pp. 929–939, 2003.
 - [20] M. Raffel, H. Richard, K. Ehrenfried et al., "Recording and evaluation methods of PIV investigations on a helicopter rotor model," *Experiments in Fluids*, vol. 36, no. 1, pp. 146–156, 2004.
 - [21] N. I. Vitzilaios and N. C. Tsourveloudis, "An experimental test bed for small unmanned helicopters," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 54, no. 5, pp. 769–794, 2009.
 - [22] Y. Nikolos, L. Doitsidis, V. Christopoulos, and N. C. Tsourveloudis, "Roll control of unmanned aerial vehicles using fuzzy logic," *WSEAS Transactions on Systems*, vol. 4, no. 2, pp. 1039–1047, 2003.
 - [23] E. N. Sanchez, H. M. Becerra, and C. M. Velez, "Combining fuzzy, PID and regulation control for an autonomous mini-helicopter," *Information Sciences*, vol. 177, no. 10, pp. 1999–2022, 2007.
 - [24] O. Cetin, S. Kurnaz, and O. Kaynak, "Fuzzy logic based approach to design of autonomous landing system for unmanned aerial vehicles," *Journal of Intelligent & Robotic Systems*, vol. 61, pp. 239–250, 2011.
 - [25] A. Bittar, N. M. F. de Oliveira, and H. V. de Figueiredo, "Hardware-in-the-loop simulation with X-plane of attitude control of a SUAV exploring atmospheric conditions," *Journal of Intelligent and Robotic Systems*, vol. 73, no. 1–4, pp. 271–287, 2014.
 - [26] H. V. Figueiredo and O. Saotome, "Simulation platform for quadricopter: using Matlab/Simulink and X-plane," in *Proceedings of the Brazilian Robotics Symposium and Latin American Robotics Symposium*, pp. 51–55, October 2012.
 - [27] S. R. B. dos Santos, C. L. Nascimento Jr., and S. N. Givigi Jr., "Experimental framework for evaluation of guidance and control algorithms for UAVs," in *Proceedings of the 21st Brazilian Congress of Mechanical Engineering*, Rio Grande do Norte, Brazil, October 2011.
 - [28] G. Padfield, *Helicopter Flight Dynamics*, Blackwell Science Ltd, 1996.
 - [29] C. Munzinger, *Development of a real-time flight simulator for an experimental model helicopter [Diploma thesis]*, Georgia Institute of Technology School of Aerospace Engineering, 1998.
 - [30] V. Gavrillets, *Autonomous aerobatic maneuvering of miniature helicopter [Ph.D. thesis]*, The MIT Press, Boston, Mass, USA, 2003.
 - [31] V. Gravilets, B. Mettler, and E. Feron, "Nonlinear Model for a Small-Size Acrobatic Helicopter," AIAA 2001-4333, 2001, <http://gewurtz.mit.edu/papers/GMF01.pdf>.
 - [32] R. Garcia and L. Barnes, "Multi-UAV simulator utilizing x-plane," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 57, no. 1–4, pp. 393–406, 2010.
 - [33] L. Research, *X-Plane Operation Manual*, 2011, http://www.x-plane.com/files/manuals/XPlane_Desktop_manual.pdf.
 - [34] S. R. Barros dos Santos, "Modeling of a hardware-in-the-loop simulator for Uav autopilot controllers," in *Proceedings of the 21st Brazilian Congress of Mechanical Engineering*, Natal, Brazil, October 2011.
 - [35] Mathworks, "Simulink Overview," 2012, <http://www.mathworks.com/products/simulink/>.
 - [36] K. Passino and S. Yurkovich, *Fuzzy Control*, Addison-Wesley, Menlo Park, Calif, USA, 1998.
 - [37] L. X. Wang, *A Course in Fuzzy Systems and Control*, Prentice Hall, Upper Saddle River, NJ, USA, 1997.