Autonomous Flight of Quadcopters in the Presence of Ground Effect

Farzad Ahmadinejad[1], Javad Bahrami[2], Mohammd Bagher Menhaj[3], Saeed Shiry Ghidary[4]
Amirkabir Robotics Center, Amirkabir University Of Technology
Tehran, Iran
[1]farzadsw@ieee.org, [2]bbahrami58@yahoo.com,{[3]menhaj, [4]shiry}@aut.ac.ir

Abstract— **Flying near the ground surface produces the ground effect which causes fluctuations in the flight dynamics of a MAV. A similar effect occurs while flying near the ceiling. In this paper, the changes on the quadcopter's dynamic due to ground effect and ceiling effect are studied, and accordingly, a controller for regulating quadcopter's altitude in these conditions is proposed. To attain autonomous flight in a GPS-denied environment, an onboard computer-vision subsystem is designed to estimate velocity and relative position of the quadcopter. A fuzzy logic controller is designed and implemented in both simulations and real-world to control the altitude of the quadcopter. The results show that this nonlinear controller has higher performance in comparison to classic controllers.**

Keywords— Quadcopter; Ground Effect; Fuzzy Logic Controller; Optical Flow.

I. Introduction

In recent years, applications of Micro Aerial Vehicles (MAVs) have introduced a new and novel research field. Aerial vehicles can be divided into two main categories based on their structure and flight capabilities: First, vertical takeoff and landing (VTOL) systems, and second, fixed wings aerial vehicles (mainly airplanes). VTOL systems are capable of flying and landing vertically, and also, hovering in fixed position mid-air. Thus, they do not need a runway or special launching equipment. One of the recent types of VTOLs is the multicopter system. Multicopters are simple in terms of structure and the only moving parts in these systems are their motors. The simple mechanical structure of these systems has made them be more durable, be cost-effective and highly customizable in comparison with other types of VTOLs, for instance, helicopters.

A quadcopter is a multicopter with 4 motors installed on a rigid structure. Because of this simple structure, quadcopters have a fast, nonlinear, and unstable dynamic. Therefore, a fast and real-time flight controller is needed to stabilize the system and control its altitude and position. There are quite a bit of research on modeling and designing the controllers for these systems. By taking into account some practical assumptions which are valid in most cases, a nonlinear mathematical model has introduced for the quadcopters [1]. Also, several linear and nonlinear control methods have been used to stabilize and control these systems and their performance has been verified both in simulation and implementation in the real world. Some of these well-known methods are PID, LQR, Feedback Linearization, and Sliding Mode. PID controller can regulate the altitude and position of the system with some limitation on agility and angles [2]. LQR controller has a better performance in the simulation in comparison to the PID controller [3]. However, it cannot be implemented easily in real-world applications, since the adjustment of the K matrix depends on the exact model of a system which is not available in most cases. In other researches, the functionality of the feedback linearization controller in comparison with adaptive sliding mode has been investigated [4]. Based on the results, feedback linearization controller is not a very good choice for implementing in the real world because of its high sensitivity to the uncertainty of the system parameters, while the sliding mode controller is more robust.

Although a lot of researches have been done on the quadcopter system, its behavior when it is flying at low altitude and with the presence of the ground effect is not studied enough. This is despite the fact that one of the main applications of VTOL MAVs is for search and rescue (SAR) operation in the confined areas and in this situation, flying near the ground surface is inevitable. For example, when searching a building in a SAR operation after an earthquake, to avoid collision with semi-closed passages, the MAV should fly through narrow corridors and near the ground surface. Pennsylvania University researchers have conducted quite a few of experiments both about ground and ceiling effect, as well as their effects on the dynamic of MAVs [5]. Based on their results, MAV's dynamic changes while flying near a surface (ground or ceiling), and this change is not negligible. It makes the amount of produced lift in constant propeller speed, changes in relation to the propeller's distance to a surface.

Altitude's control for terrain following applications has been studied by B.Hérissé et al. [6]. In this research, the flying altitude of the MAV was 1 meter above ground, and as a result, the ground effect did not factor in the experiments. On the other hand, the controlling algorithm which was postulated in this experiment works when the speed of the MAV is measurable, which is not always feasible.

To achieve autonomous flight for a quadcopter, real-time position feedback is needed for the control loop. The position of the robot can be measured in two methods. The first method is using a global measurement system, located outside of the robot. For example, a commercial Global Positioning System (GPS) can estimate the position of the robot in open air spaces with an

978-1-7281-1194-0/18/$31.00 ©2018 IEEE

error of few meters [7] and a pose capturing system using a set of cameras can locate the robot in a room with high accuracy [8]. These methods are not suitable for positioning the robot in an unknown and confined environment which is common in SAR operations. On the other hand, in the second method, the position of a robot is estimated using only its onboard sensors and therefore it is more versatile for indoor navigation. Most of the quadcopters are equipped with a small camera and some processor unit. Therefore, computer vision algorithms can be applied to the image data for quadcopter's localization. Optical flow is one of the most used computer vision algorithms to estimate the position of quadcopters [9]. It provides a relative velocity of the robot to its environment (mostly the ground surface) and by integrating the velocity, position of the robot can be estimated.

In this paper, the quadcopter system is introduced in section 2, then, the ground effect and its effect on the dynamic of the system is studied. The fuzzy controller for controlling the altitude of the system is introduced in section 3, and a method for estimating the altitude and velocity of MAV in relative to the ground using optical-flow algorithm is explained in section 4. In section 5, the performance of the proposed controller is studied in both simulations and the actual system.

II. Quadcopter System and Ground Effect

A. The mathematical model of a quadcopter system

The quadcopter's structure is made of a rigid structure with four arms and the actuators are installed at the end of these arms. To minimize rotational inertia, the other pieces of equipment are installed close to the center of the structure. The actuators are consist of motors, propellers, and electrical speed controllers (motor's driver), in a way that the central controller can set the rotational speed of propellers by sending a corresponding signal. As shown in fig. 1, each actuator causes a lifting force proportional to the speed of the blade. Therefore, resultant lifting force's (F) direction is in the opposite direction of quadcopter frame's z (e3b) axis, and the only its magnitude can change.

Thus, if the angle of the frame does not change in relative to the ground frame, increasing and decreasing of the F will cause an acceleration in the direction of ground's z-axis (altitude). For maneuver of the system in 3D space, the angle of the MAV changes in relative to ground, therefore, the F will have a horizontal component, resulting the system to accelerate in the horizontal direction. Rotation of MAV's frame is achieved by changing the lifting force of each of actuator individually. For instance, by increasing F1 force and reducing F3 force the frame of the MAV will rotate around y (e2b) axis, because of created torque.

The propellers which are used in this system are in two types of clockwise (puller) and counterclockwise (pusher). To produce the lifting force, two motors run in a clockwise direction and two motors in a counterclockwise direction. As a result, if the speed of the motors is equal, the resultant torque around z-axis is neutralized, and the MAV doesn't rotate around the z-axis (Yaw rotation). For changing Yaw angle, the speed of motors which are facing each other should be changed.

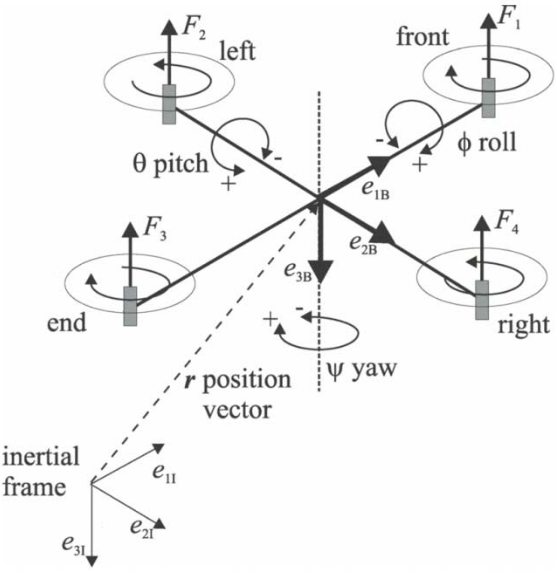

Fig. 1. A quadcopter frame diagram

It should be noted that z-axis in this system is in the downward direction. The reason for that is to comply with coordinates of NED which is accepted as a standard in the aviation field. Also, the final output of each actuator is its lifting force not the speed of the blades.

Like any other rigid structure in the space, the orientation of this system can be determined by three Euler angle: roll (φ), pitch (θ), and yaw (ψ). These three angles create attitude vector Ω.

$$\Omega^T = (\phi, \theta, \psi) \tag{1}$$

Position of the MAV is determined by vector R:

$$R^T = (x, y, z) \tag{2}$$

Also, the rotation of MAV's frame in relative to the reference frame is represented by a rotation matrix in (3). In this equation, c_θ, s_θ represent respectively $\cos\theta, \sin\theta$:

$$R = \begin{pmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{pmatrix} \tag{3}$$

The relation between the rotation speed of each motor and the lifting force is represented in (4). F is the lift force, ω represents the rotation speed of the motor and b is a constant which depends on the dimensions and aerodynamic characteristics of the propeller.

$$F_i = b.\omega_i^2 \tag{4}$$

In a real system, the rotational speed of motors is used as a controlling input. Nevertheless, we define new controlling inputs "u" for the system to simplify the modeling. Equation 8 shows the relation between new inputs and rotation speed of motors. This relation is represented by an invertible transpose matrix. Hence, the value of motors rotational speed can be calculated from u.

$$\begin{aligned} u_1 &= b(\omega_1^2+\omega_2^2+\omega_3^2+\omega_4^2) \\ u_2 &= b(\omega_2^2-\omega_4^2) \\ u_3 &= b(\omega_1^2-\omega_3^2) \\ u_4 &= d(\omega_1^2-\omega_2^2+\omega_3^2-\omega_4^2) \end{aligned} \tag{5}$$

As a result, gyroscopic torque which is dependent to the rotation speed of a motor is defined as a function of u:

$$g(\underline{u}) = \omega_1-\omega_2+\omega_3-\omega_4 \tag{6}$$

By defining new inputs, the dynamical model of the system will be [1]:

$$\begin{aligned} \ddot{x} &= -(\cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi).\frac{u_1}{m} \\ \ddot{y} &= -(\cos\phi\sin\theta\sin\psi-\sin\phi\cos\psi).\frac{u_1}{m} \\ \ddot{z} &= g-(\cos\phi\cos\theta).\frac{u_1}{m} \\ \ddot{\phi} &= \dot{\theta}\dot{\psi}(\frac{I_y-I_z}{I_x})-\frac{I_R}{I_x}\dot{\theta}.g(u)+\frac{L}{I_x}u_2 \\ \ddot{\theta} &= \dot{\phi}\dot{\psi}(\frac{I_z-I_x}{I_y})+\frac{I_R}{I_y}\dot{\phi}.g(u)+\frac{L}{I_y}u_3 \\ \ddot{\psi} &= \dot{\theta}\dot{\phi}(\frac{I_x-I_y}{I_z})+\frac{1}{I_z}u_4 \end{aligned} \tag{7}$$

The value of parameters in this dynamical model for an actual quadcopter is in SolidWorks software. Knowing these parameters are crucial for simulation and controller design. These parameters are shown in table 1:

TABLE I. PARAMETERS IN A QUADCOPTER MODEL

Parameter	*Value*	*Unit*
G	9.81	$\frac{m}{s^2}$
M	0.9	kg
L	0.25	m
$I_x = I_y$	5.42 x 10⁻³	kgm^2
I_z	9.37 x 10⁻⁵	kgm^2
I_R	4.13 x 10⁻⁵	kgm^2

B. The ground and ceiling effect

Ground effect phenomenon occurs when the flying system is near the ground surface. This phenomenon causes the changes in the relationship between the speed of propeller and amount of the lifting force. In other words, b parameter in (4) is not a constant parameter, and it has different values depending on the altitude of the flight.

The ground effect causes that near the surface of the ground, despite the same rotating speed, a more lifting force is generated (the value of b parameter is increased). Because of complicated aerodynamic relations, there is not an exact model for ground effect. Nevertheless, the relation between ground effect and altitude can be achieved by an experiment [5].

To estimate the relation between ground effect and altitude, the required rotation speed for producing a constant lifting force at different levels of altitude is measured. In the experiment, a 4.5 x 10 propeller was used. Measured values for the required speed to provide 10 N lifting force at the different level of altitude are shown in fig. 2. Based on this curve, the influence of ground effect at low altitude is considerable. However, at an altitude of higher than 0.5 meters, this effect can be neglected. As a general rule to be considered, it can be said that the ground effect at altitudes more than 2 times of blades diameter could be neglected.

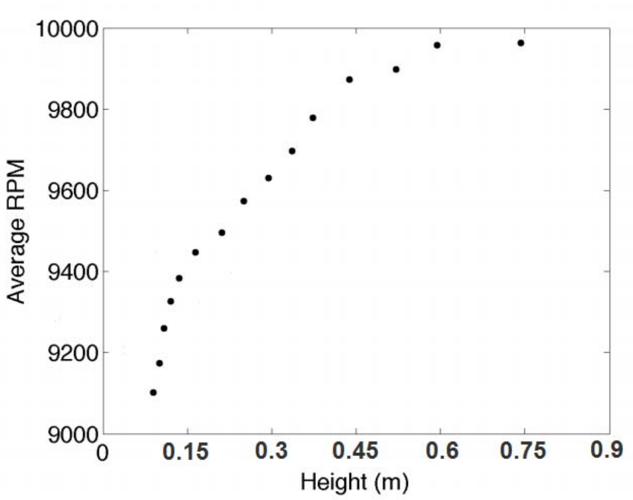

Fig. 2. Ground effect in different altitudes

Based on this experiment, an approximate mathematical model for ground effect is introduced in (8). In this equation, T expresses the amount of lifting force throughout the flight at an altitude of z, the amount of lifting force during the flight at very high altitude, and D is the diameter of the blade. It should be noted that this equation is valid for altitudes more than D/2:

$$\frac{T}{T_\infty} = \frac{1}{1-\left(\frac{D}{4z}\right)^2} \tag{8}$$

By applying this mathematical model to (7), a more precise model for transitional dynamics of quadcopter in the presence of ground effect could be achieved.

$$\ddot{x} = -(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi).\frac{u_1}{m}.\frac{1}{1-\left(\frac{D}{4z}\right)^2}$$
$$\ddot{y} = -(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi).\frac{u_1}{m}.\frac{1}{1-\left(\frac{D}{4z}\right)^2} \quad (9)$$
$$\ddot{z} = g - (\cos\phi\cos\theta).\frac{u_1}{m}.\frac{1}{1-\left(\frac{D}{4z}\right)^2}$$

One of the MAV's applications is passing on the obstacles at low altitudes (canals, etc.). In these situations, the distance between the MAV and the roof above it is low and the ceiling effect will appear [5]. This effect has a similar characteristic to ground effect, and it can be approximated to Eq. (8). The major difference between this phenomenon and the ground effect is in how they influence the dynamic of the MAV system. Both of these phenomena cause an increase in the lifting force (for the same rotating speed of the propeller), and therefore, they cause the MAV to climb. In the ground effect, by climbing the MAV the influence of ground effect decreases, on the other hand, in the ceiling effect, this effect escalates. In other words, the ground effect has a negative feedback on the dynamics of the flight, but ceiling effect has positive feedback on that and it is possible to make the whole system unstable.

III. Controller Design

In reality, it can be said that the dynamic of the rotational movement of a quadcopter is much faster than the dynamic of its transitional movement; therefore, the controllers are designed in two stages. The first stage is related to the controlling the position of MAV, and the outputs of this stage are desired roll, pitch, and the controlling signal (u). Afterward, these angles are used as reference inputs of the next stage of the controller. Figure 5 shows the block diagram of this controller.

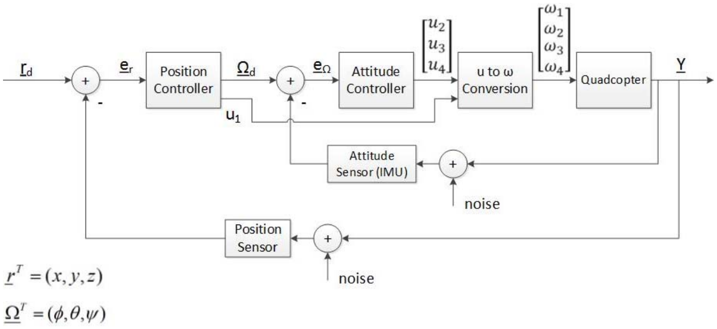
$\underline{r}_d$
+
e_r
Position Controller
$\underline{\Omega}_d$
+
e_Ω
Attitude Controller
$\begin{bmatrix}u_2\\u_3\\u_4\end{bmatrix}$
u to ω Conversion
$\begin{bmatrix}\omega_1\\\omega_2\\\omega_3\\\omega_4\end{bmatrix}$
Quadcopter
Y
-
u_1
-
Attitude Sensor (IMU)
+
noise
Position Sensor
+
noise
$\underline{r}^T = (x, y, z)$
$\underline{\Omega}^T = (\phi, \theta, \psi)$

Fig. 3. The diagram of the two stage controller

A. Controlling the angles (attitude)

Considering the fact that the system's behavior has a nonlinear nature, for achieving a complete tracking, a nonlinear and suitable controller is required. For designing a nonlinear controller, feedback linearization method is used.

In the case that inertia of motors is a small value (like the real situation), the gyroscopic effect is negligible and the system model will look like the below:

$$\ddot{\phi} = \dot{\theta}\dot{\psi}(\frac{I_y - I_z}{I_x}) + \frac{L}{I_x}u_2$$
$$\ddot{\theta} = \dot{\phi}\dot{\psi}(\frac{I_z - I_x}{I_y}) + \frac{L}{I_y}u_3 \quad (10)$$
$$\ddot{\psi} = \dot{\theta}\dot{\phi}(\frac{I_x - I_y}{I_z}) + \frac{1}{I_z}u_4$$

Assuming the states of the system are:

$$x^T = \begin{bmatrix} x_1\\x_2\\x_3\\x_4\\x_5\\x_6\\x_7\\x_8\\x_9 \end{bmatrix} = \begin{bmatrix} \dot{x}\\\dot{y}\\\dot{z}\\\phi\\\theta\\\psi\\\dot{\phi}\\\dot{\theta}\\\dot{\psi} \end{bmatrix} \quad (11)$$

Using the feedback linearization method [10], control law for stabilizing the attitude of the system will be:

$$u_2 = w_2(x_{4d} - x_4) + \frac{I_x}{L}\left(K_2x_7 - x_8x_9\,(\frac{I_y - I_z}{I_x})\right)$$
$$u_3 = w_3(x_{5d} - x_5) + \frac{I_y}{L}\left(K_3x_8 - x_7x_9\,(\frac{I_z - I_x}{I_y})\right) \quad (12)$$
$$u_4 = w_4(x_{6d} - x_6) + I_z\left(K_4x_9 - x_7x_8\,(\frac{I_x - I_y}{I_z})\right)$$

In (12), K_i and w_i are control parameters that can change the response of the system. By setting K_2, K_3, K_4 a negative value, the stability of quadcopter's attitude is guaranteed.

B. Position Control

System's position control consists of horizontal position and altitude control. Horizontal control is feasible by changing the angle of the system. A method for measuring required angles for transitional movement by using linearized feedback method is proposed in [10]. On the other hand, altitude's control consists of many uncertainties, especially, the mass of system and ground effect. Because of dependency on the environment's conditions, exact modeling of ground effect is not feasible, and as a result, feedback linearization method does not work properly. In this situation which the exact model is not available, fuzzy logic controllers (FLC) can be used.

There are several methods of designing a fuzzy inference system. Among these methods, Mamdani and Takagi Sugeno are the most common ones. In this paper, a Mamdani fuzzy logic controller method has been used for controlling the altitude of the robot and other states of the system are controlled by a feedback linearization method. In this altitude controller, proportional error and its derivative are used as an input of a fuzzifier and are converted to the fuzzy variables. Then, the inference system determines the output fuzzy variable by considering some predefined conditional rules. For use in the controlling systems, output fuzzy variable is converted to a number by a defuzzifier. Actually this method works like a PD controller; however, this controller can have some characteristics of a non-linear controller based on defined rules.

For designing the fuzzy controller, Matlab has been used as the simulation environment. This controller has two inputs (altitude's error and its derivative) and one output (amount of lifting force). Figure 4 illustrates membership functions for input and output. The top left figure is the membership function of altitude's error which divides into five subsets of very negative, negative, zero, positive, and very positive. These membership functions range from -40cm to 40cm, and the fuzzy variable is saturated out of this range. The top right figure shows the input membership of error derivative. It has three subsets of decreasing (negative speed), constant (zero speed), and increasing (positive speed). The down left figure shows membership function of the system's fuzzy output (lifting force), and it has 5 regions range from very low to very high. In the last step of the fuzzy control, the amount of fuzzy output which are available as fuzzy variables, are converted to the numbers by a centroid defuzzifier.

In this regards, input information is converted to the fuzzy variables, and it is possible to define fuzzy rules for them. Rules which are used in this controller are according to table 2. For example, if the error has a big negative value and error derivative is negative too (the error is increasing), lifting force will be increased dramatically, and as a result, amount of error decreases. These rules are defined based on system's behavior in experiments.

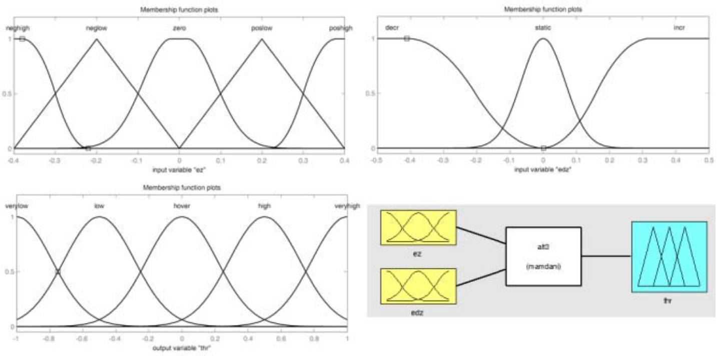

Fig. 4. The membership function of the fuzzy controller's input and output

TABLE II. FUZZY RULES

		Error				
		Very Negative	***Negative***	***Zero***	***Positive***	***Very Positive***
Derivitive of Error	***Decreasing***	Very High	High	High	Average	Low
	Constant	High	High	Average	Low	Low
	Increasing	Average	Average	Low	Low	Very Low

IV. POSITION ESTIMATION

Optical flow is one of the main methods in computer vision to estimate displacement from an image sequence. One of the most important assumptions in this algorithm is the brightness constancy constraints, which assumes that image brightness changes between frames are due only to camera or object motion. Nevertheless, implementing this algorithm on every pixel of a large image is a time-consuming process and it prevents optical flow algorithms from being used in most real-time applications.

In recent years, a number of different implementations have been introduced for the optical flow algorithm. The basic approach for these works is among hardware implementation and parallel processing techniques. GPGPUs are one of the best candidates for implementing optical flow algorithm, however, they have high power consumption in comparison with FPGAs and ASICs. Moreover, the considerable size and weight of GPGPUs have prevented them to be a suitable choice for being used in MAVs. ASIC design could be a much more efficient solution, nevertheless, it has a high cost and time to market. As a result, among all the choices, FPGAs are the best candidate. In comparison with ASICs, these ICs have a number of benefits; low cost, high flexibility, low time to market, low power consumption, to name but a few.

Many optical flow algorithms have been introduced in recent years. Optical flow algorithm can be implemented in two General methods: sparse and dense optical flow algorithm. In the former, the movement between the two frames is measured by tracking a set of key features in the images. In the latter, the algorithm is designed to work on all, or subsets of pixels. The advantage of dense optical flow algorithm is that it is much more accurate, however, it requires more computation.

Currently, 3D tensor techniques can produce dense and accurate dense optical flow fields [9]. There are different types of 3D tensors based on different formulations. The gradient tensor is used in our design because it is easier to implement with pipelined hardware. In this work, a tensor-based optical flow algorithm is used and implemented by a pipeline architecture on an FPGA for achieving 60fps.

By using optical flow algorithm, the speed of the vehicle's movement relative to the ground can be measured, and consequently, by integrating the velocity vector of the robot, the position of the MAV can be estimated. Integrating the velocity vector causes that error increase over time, and as a result, the estimation will not be valid for a long period of time. Although there are some errors in positioning of the MAV in long term, it is accurate enough in a short period of time. To reduce integration error, an estimated position will be compared with a reference landmark periodically throughout the movement of the MAV.

V. Simulation and Implementation Results

In this work, the performance of the proposed controller in presence of Ground effect is evaluated in both simulation and real-world experiment.

A. Simulation Results

The simulations have been designed in the Matlab/ Simulink software. As the first simulation, the performance of the proposed altitude controller is tested without Ground effect. Figure 5 shows the result of the simulation. It can be seen that the Fuzzy controller can track the desired altitude and also F.L. controller can stabilize attitude and horizontal position of the quadcopter.

In the next simulation, the Ground effect and Ceiling effect was added to the quadcopter model. Performance of the Fuzzy controller is compared to a conventional PID controller in this situation. Figure 6 shows the influence of Ground/Ceiling effect on altitude control of the quadcopter. Please note that in the simulations z-axis is in the downward direction and therefore, higher altitude value means a lower distance from the ground.

The ground effect act as an external disturbance and only causes some error in tracking when the quadcopter gests near the ground. On the other hand, ceiling effect can make the system with PID controller unstable. In both cases, the fuzzy controller has better performance with a small error around the reference value (Fig. 7).

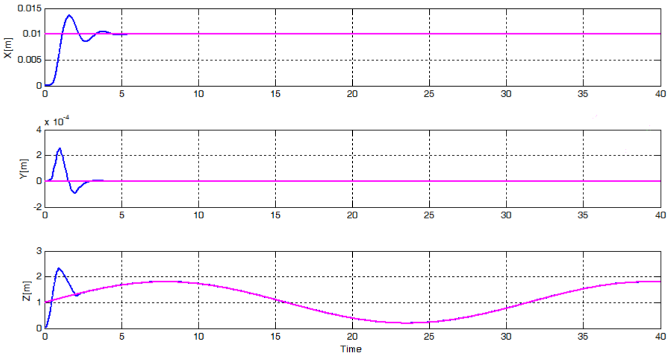

Fig. 5. Simulation result of the controller without Ground effect

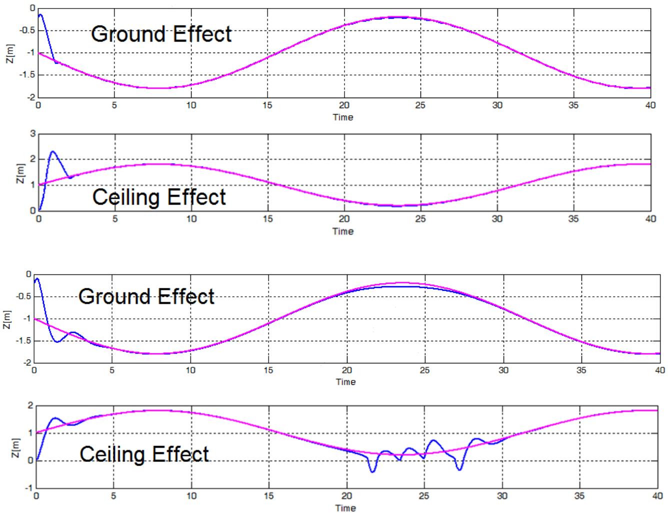

Fig. 6. Top, Altitude control using Fuzzy logic controller. Bottom, Altitude control using PID controller

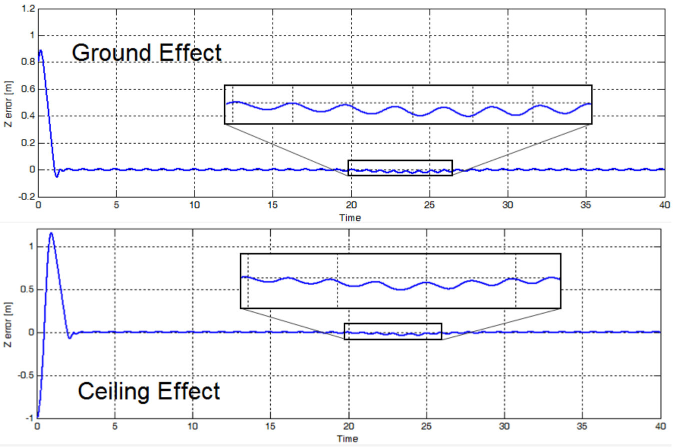

Fig. 7. Error of Fuzzy logic controller for altitude control

B. Implementation Resuls

A quadcopter system was designed and built for this project (Fig. 8). A Pixhawk flight controller is used as the main board and the controller software was programmed to it. The attitude of the system is measured by MEMS IMUs. A laser distance sensor is used as altitude's measurement sensors.

For measuring the horizontal position of the MAV, one camera is installed under the robot which films the surface under the robot. The movement speed of the quadcopter is estimated by calculating Optical Flow of the downward camera. The image processing calculations are performed in 60 frames per second to reduce the latency.

In the first experiment, the performance of the fuzzy controller for maintaining a small distance (20 cm) over the ground is tested. Figure 9 shows the plot of recorded sensor data.

In the second experiment, the quadcopter was set to follow a 30-degree ramp autonomously. The proposed control system can follow the ramp with 1 m/s velocity while maintaining 30cm distance over the terrain.

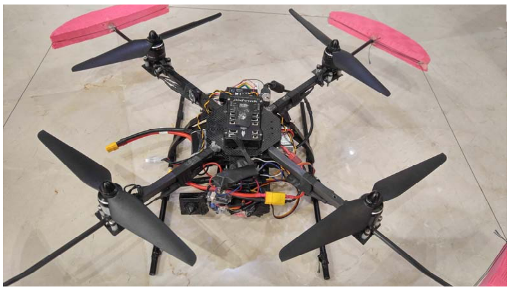

Fig. 8. A picture of the custom-built quadcopter

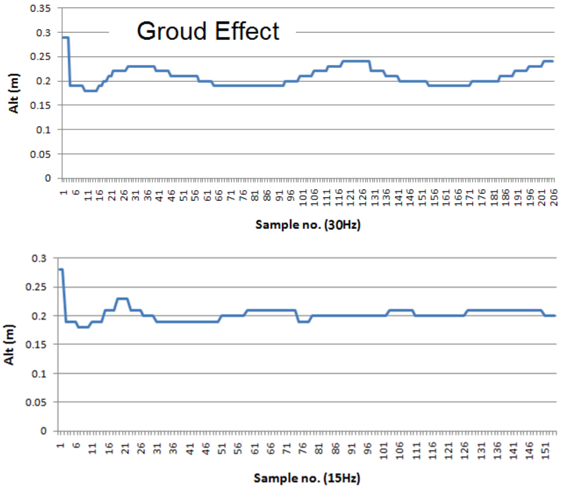

Fig. 9. The result of altitude control on the actual robot. Top, PID controller, bottom Fuzzy logic controller

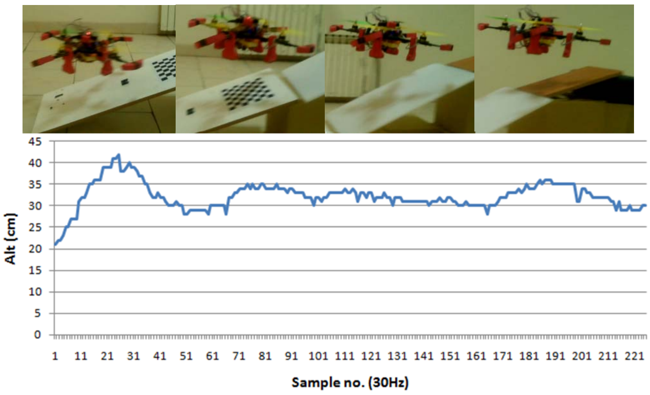

Fig. 10. Following a ramp autonomously with a constant altitude

VI. CONCLUSIONS AND FUTURE WORKS

This paper presents a nonlinear controller system for the autonomous flight of a quadcopter. This controller performs well in low altitude flight and near ground/ceiling surfaces. The Optical flow algorithm was used to estimate the relative position of the robot by processing system's camera in real-time using an onboard processing unit. It is shown that the feedback linearization algorithm can stabilize the attitude and horizontal position of the quadcopter and the Fuzzy logic controller is a robust controller for altitude control of quadcopter in presence of ground effect especially if compared with a simple PID controller.

For future works, a SLAM algorithm can be used to navigate the quadcopter in environment enable the robot to explore a multistorey building. Also apart from simulation, the actual performance of system near ceiling surface needs to be studied. Flying a narrow passage like a tunnel would be an appropriate scenario for testing ceiling effect on flight performance of the system.

REFERENCES

[1] K.Nonami, F.Kendoul, S.Suzuki, W.Wang and D.Nakazawa, "Mathematical Modeling and Nonlinear Control of VTOL Aerial Vehicles," Autonomous Flying Robots,2010, Part II, Pages 161-193.

[2] P. Wang, Z. Man, Z. Cao, J. Zheng and Y. Zhao, "Dynamics modelling and linear control of quadcopter," 2016 International Conference on Advanced Mechatronic Systems (ICAMechS), Melbourne, VIC, 2016, pp. 498-503.

[3] L. M. Argentim, W. C. Rezende, P. E. Santos and R. A. Aguiar, "PID, LQR and LQR-PID on a quadcopter platform," 2013 International Conference on Informatics, Electronics and Vision (ICIEV), Dhaka, 2013, pp. 1-6.

[4] D. Lee, H.J. Kim, Sh. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," Int. J. Control Autom. Syst. (2009), Volume 7, Issue 3, pp 419–428.

[5] Caitlin Powers, Daniel Mellinger, Alex Kushleyev, Bruce Kothmann, and Vijay Kumar. "Influence of Aerodynamics and Proximity Effects in Quadrotor Flight." Int. Symposium on Experimental Robotics, June 2012

[6] B.Hérissé, T.Hamel, R.Mahony and F.X.Russotto, "A terrain-following control approach for a VTOL Unmanned Aerial Vehicle using average optical flow," Autonomous Robots, 2010, Volume 29, Numbers 3-4, Pages 381-399.

[7] U. Iqbal, J. Georgy, W. F. Abdelfatah, M. J. Korenberg and A. Noureldin, "Pseudoranges Error Correction in Partial GPS Outages for a Nonlinear Tightly Coupled Integrated System," in IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 3, pp. 1510-1525, Sept. 2013.

[8] F. Ahmadinejad, H. Farazi and S. Shiry Ghidary, "A low-cost vision-based tracking system for position control of quadrotor robots," 2013 First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM), Tehran, 2013, pp. 356-361.

[9] Z. W. Zhaoyi, D. J. Lee and B. E. Nelson, "FPGA-based Real-time Optical Flow Algorithm Design and Implementation," Journal of Multimedia 2 (2007): 38-45.

[10] H. Voos, "Nonlinear control of a quadrotor micro-UAV using feedback-linearization," 2009 IEEE International Conference on Mechatronics, Malaga, 2009, pp. 1-6.