

## Autopilot Design of Unmanned Aerial Vehicle

Ahmed Hassan Ahmed\*, Bahaaeldin Gamal, Ahmed Nasr Ouda, Ahmed Mohsen Kamel and Yehia Zakaria El-Halwagy

Military Technical Research Center, Cairo, Egypt

\*Corresponding author: Ahmed AH, Military Technical Research Center, Cairo, Egypt, Tel: +2-02-24036110; E-mail: Ahmed\_hassan\_aircraft@yahoo.com

Received: April 17, 2018; Accepted: April 28, 2018; Published: May 02, 2018

Copyright: © 2018 Ahmed AH, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

### Abstract

In this paper, a complete system of quad rotor stability mechanism was designed and implemented. Starting with a Single-axis Implementation of a Quad rotor, SISO approach is implemented for control structure to achieve desired objectives. The tradition PID, modified PID controllers and Adaptive Neural fuzzy inference system (ANFIS) were executed on simulation model of Quadrotor. The performance of the designed control structure is evaluated through processor in loop experiment and through the time domain factors such as overshoot, settling time. The evaluation results reveal that the designed modified PID controller has the best performance, more robust, higher stability and less control effort compared to the original and designed ANFIS controller. Three axis test and implementation of the algorithm has been performed and the results are discussed.

**Keywords:** Single-Input Single-Output (SISO); Proportional-Integral-Derivative (PID); Adaptive Neural Fuzzy Inference System (ANFIS)

### Introduction

Recent technological advances in energy storage devices, sensors, actuators and information processing have boosted the development of Unmanned Aerial Vehicle (UAV) platforms with significant mission capabilities [1]. Unmanned aerial vehicles are important when it comes to perform a desired task in a dangerous and/or inaccessible environment. More recently, a growing interest in un-manned aerial vehicles (UAVs) has been shown among the research community [2]. The rotorcraft UAVs pose a set of advantages compared to the fixed wing UAVs, such as hovering, vertical takeoff and landing and an aggressive maneuvering. Within the family of the rotor-crafts, Unmanned Quadrotor Helicopters (UQHs) have gained increasing attention among scientists and engineers [3].

### Materials and Methods

A quadrotor is a 4-rotor vertical take-off and landing vehicle that has the maneuvering abilities of traditional helicopters with significantly lower mechanical complexity. This low complexity increases dependability while reducing the cost of manufacturing, operation, and maintenance [4]. The basic quad-rotor design consists of four complete rotor assemblies attached at equal distances from each other and a central hub. The quad-rotor is an under-actuated, dynamic vehicle with four control inputs and six degrees of freedom [5].

All the rotors are located within the same plane and oriented such that the thrust generated by each rotor is perpendicular to the vehicle. Quadrotors use 2 sets of identical fixed pitched propellers, 2 clockwise (CW) and 2 counter-clockwise (CCW) which nullifies torque produced by rotors and eliminates the need of tail rotor used in conventional helicopters. The movement for this aircraft is achieved by manipulating the thrust produced by individual propeller.

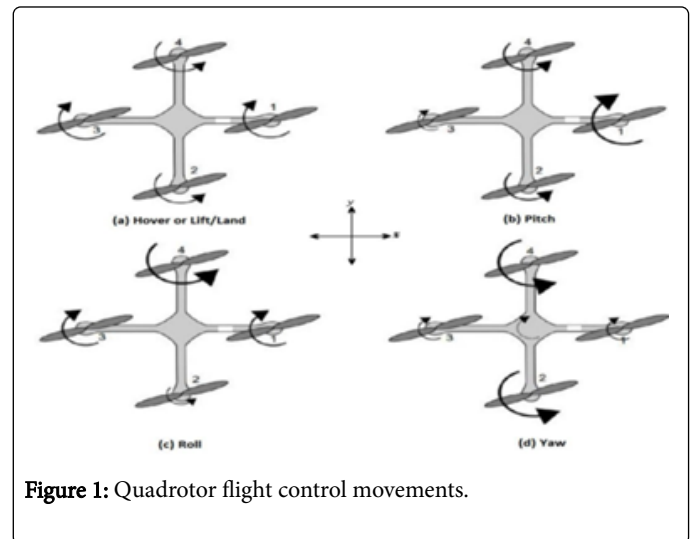


Figure 1: Quadrotor flight control movements.

Figure 1 illustrates basic flight control movement for quadrotor. Assume motor is front of aircraft and sizes of arrows represent the magnitude of thrust produced by corresponding motor. Some components, such as multi-levelled equations, graphics, and tables are not prescribed, although the various table text styles are provided. The formatter will need to create these components, incorporating the applicable criteria that follow:

- **Hover or Lift/Land:** To make the aircraft hover at particular altitude all four rotors are kept at same speed. To lift or land the aircraft, speed of all four rotors is increased or decreased by equal amount respectively.
- **Pitch:** For pitch movement around y-axis, speed of motor 1 and 3 is changed.
- **Roll:** For roll movement around x-axis, speed of motor 2 and 4 is changed.

- **Yaw:** For yaw movement around z-axis, torque difference is produced by changing speed of pair of motors on x-and y-axis.

This paper is divided as follows: Section II provides the system architecture that describes the basic parts of the entire system while Section III discusses the software design and shows the flowchart of the program section IV, V show the implementation of the designed control and altitude hold. Finally, Section VI concludes the paper conclusion.

### Traditional controller design

The control performs a key role in the quad-rotor's stability, making possible to control precisely the attitude and altitude states of the aircraft. Its main goal is to make the quad-rotor moves to a new desired position (called reference) and react to external disturbances quickly and in a controlled way. Attitude control is the key element to maintain stability during flight. There were two different PID controllers that used in the experiment. The first trail was design position control only without speed control, it found that the system response was slow and not achieved the requirements so both position and speed controllers are implemented on this system to enhance its performance.

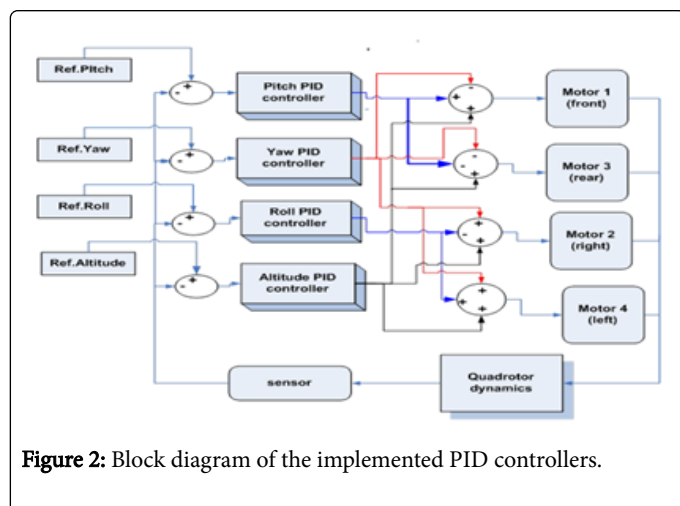


Figure 2: Block diagram of the implemented PID controllers.

Figure 2 illustrates these four implemented controllers along with the entire system. For the pitch angle, the controllers output modifies the speed of the front and rear motors, whereas in front its value is increased and in the rear its value is decreased. For the roll angle the controllers output modifies the speed of the left and right motors, whereas in the right its value is increased and in the left its value is decreased. For the yaw angle the controllers output modified the speed of the four motors, whereas in the front and rear its values are increased and in the right and left its values are decreased. The resulting rotation moment is since the pitch of the front-rear propellers is opposite to the right-left propellers. For altitude, the controllers output modifies the four motors simultaneously, being increased its values to ascend and decreased to descend the UAV [6].

### The modified PID controller

The industrial application used liner PID regulators due to its simple structure, good performance for several processes and tunable even without a specific model of the controlled system [7].

The traditional PID structure presents main drawbacks:

- The derivate action is calculated from the error. If the task adds a step in the reference, the output of the derivate would present an impulse. This sharp movement can saturate the actuators and push away the system from the linear zone. For this reason, most of the PID architecture presents the derivative action of the process output only.
- The desired angle to the controller is a step input or abrupt change then, because of the presence of the derivative term in the control action, the output  $u(t)$  will not involve an impulse function but will involve a sharp pulse function. Such a phenomenon is called set-point kick [8]. To avoid the set-point kick phenomenon, the operation of the derivative action will be only in the feedback path so that differentiation occurs only on the feedback signal and not on the reference signal.

The control scheme arranged in this way is called the modified PID control as shown in Figure 3.

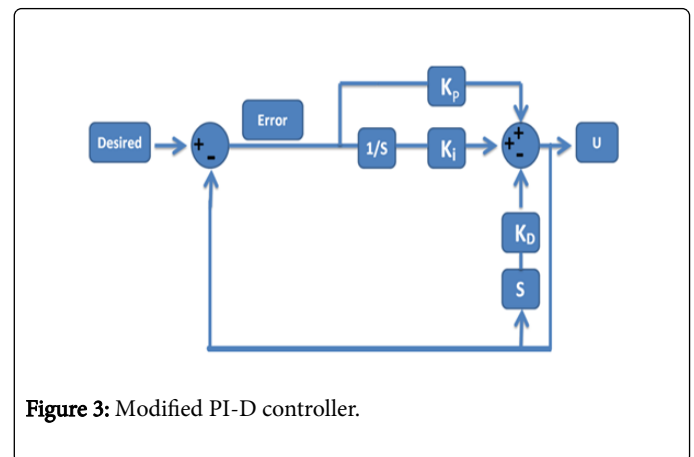


Figure 3: Modified PI-D controller.

### The implementation of the designed controllers

As shown in previous section the PID controller and Modified PID controllers were designed. In this section a PID controller and Modified PID were designed and implemented on ARDUINO board using C++ program.

**PID implementation on one axis:** In Figure 4, each channel was tested separately; a lot of trials were done to tune the designed PID controller. During the tuning Ziegler–Nichols second method was used.

**Implementation of the modified PID on one axis:** In the following test the one channel of Quadcopter is subjected it's supposed to reject that disturbance and maintain about zero situation.

This disturbance is done by hitting the Quadcopter and observes its response while trying to maintain zero situations [9].

In Figure 5, modified PID in software design consists of two loops, inner loop which called speed controller which depends on angle's rate and the other one called angular position which depends on angle's value. The tuning of this control starts from the inner loop gains and then adjusts the outer loop's gains.

The next table shows the results of the designed PID and modified PID controllers. The experiment is conducted by going through the following steps:

- Connect the two speed controllers to the two motors and the control signal of the motors is connected to ARDUINO board.
- Connect the IMU to main ARDUINO as shown.
- Connect the power as shown to all the circuit.
- After running the quad check the rotation of the motors if there is any of them rotate in the unrequired direction then invert the side wires of the three wires of the motors.

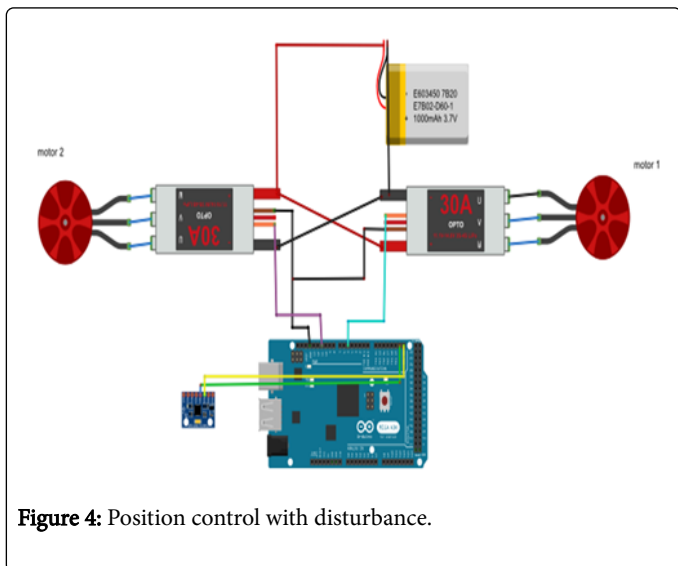


Figure 4: Position control with disturbance.

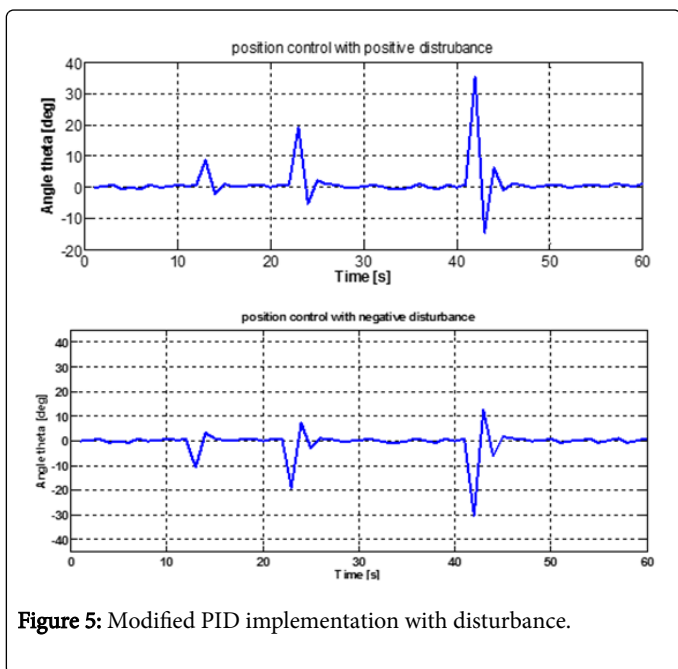


Figure 5: Modified PID implementation with disturbance.

The purpose of the designed PID controller is to maintain the one channel at zero, so the input to the PID controller is the real angles and the set point is zero position as shown in Figure 6.

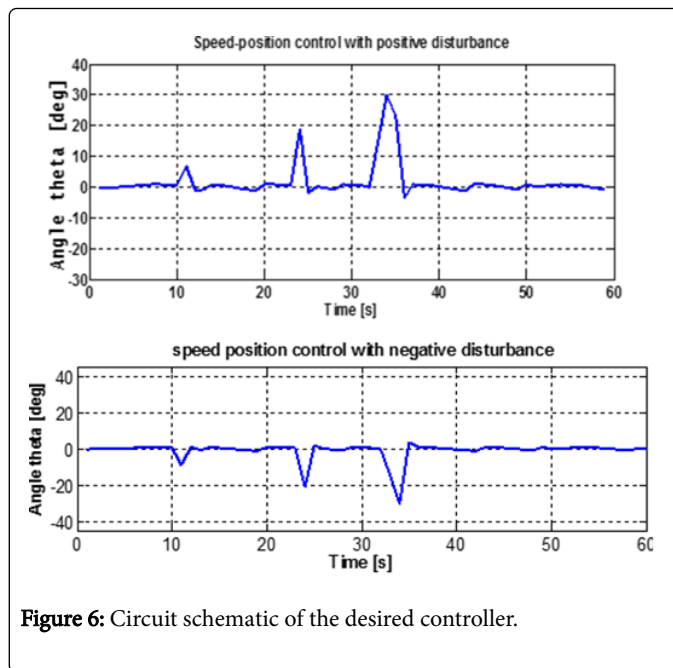


Figure 6: Circuit schematic of the desired controller.

Variables	Rise time(s)	Settling time(s)	Steady state error
PID	1.1	4	0
Modified PID	0.8	2	0

Table 1: Results of the designed PID and modified PID controllers.

Table 1 shows the output of the designed PID controller after applying disturbance .it can be rejected but with high oscillation. The output of the designed Modified PID controller with applying disturbance can be rejected but with lower oscillation than PID.

**Adaptive Neuro-fuzzy Logic Controller (ANFIS) controller:** In this section, as the system is nonlinear ANFIS can be used as a non-linear controller.

The input–output data of the PID controlled system at different operating points are used as training and checking data sets for ANFIS. Half of these data are used as a training data and the other half as a checking data. Training the FIS is started using hybrid optimization method after generating the initial FIS structure. In the case studied, a sixty training Epochs with zero tolerance error are used in the system training. Validation of the model obtained is performed then the generated FIS is successfully used as a controller of the system in the Simulink model. The ANFIS model structure is the same in all channels and can be seen in Figure 7. The surface viewer of the FIS that relates the inputs and the output of roll, pitch, yaw, and altitude channels can be seen in Figure 8.

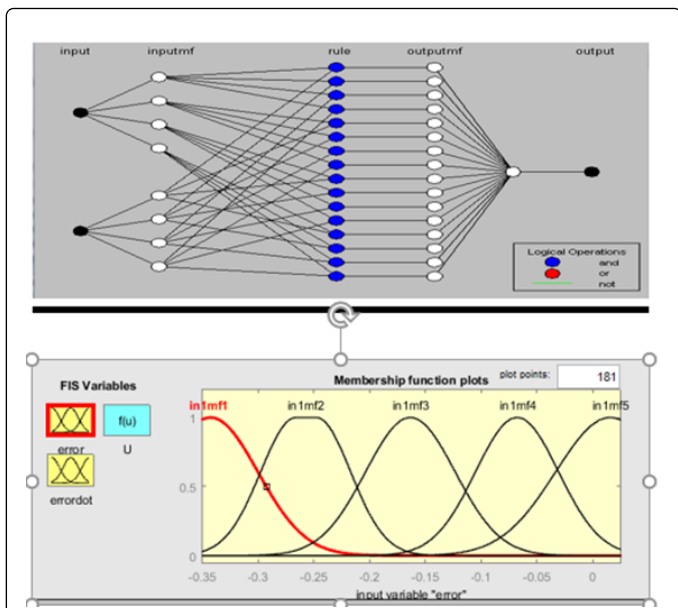


Figure 7: The ANFIS model structure.

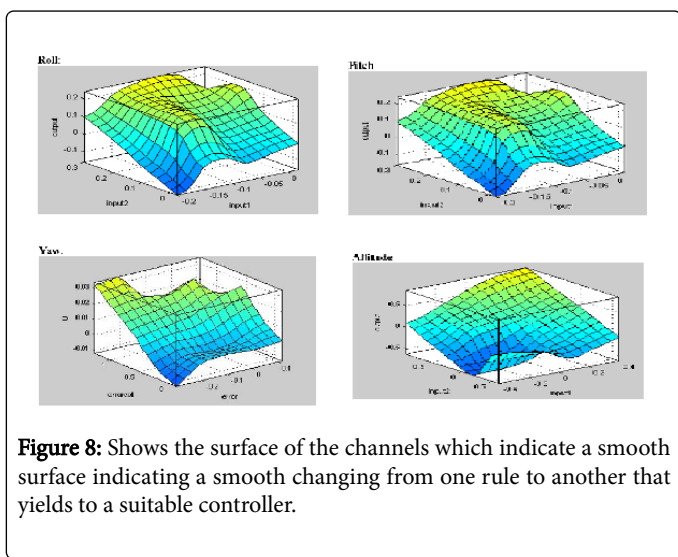


Figure 8: Shows the surface of the channels which indicate a smooth surface indicating a smooth changing from one rule to another that yields to a suitable controller.

### Designed ANFIS controller hovering stabilization

The desired parameters are:  $[X; Y; Z, \psi]=[0, 0, 3, 0]$  where the quadcopter is to reach the altitude of 3 m and taking into consideration that there is initial error in roll and pitch 0.3 rad and yaw angles of 0.5 rad. The obtained results are shown in Figure 9.

As shown in the previous figure, ANFIS controller hovering stabilization roll response has a settling time about 7 sec also; it has a low overshoot about 0.06 rad with oscillations until reach the required value.

The pitch response has a settling time about 4 sec also; it has a high overshoot about 0.3 rad with low oscillations until reach the required value. The yaw response has a settling time about 2 sec also; it has no overshoot with low steady state error about 0.003 rad. The altitude

response has a rise time about 2.7 sec also; it has no overshoot with steady state error about 5 cm.

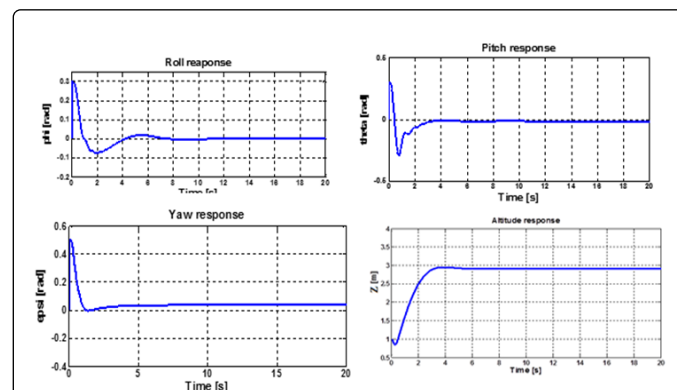


Figure 9: Hovering stabilization using ANFIS controller.

ANFIS controller hovering stabilization results are shown in Table 2.

Variables	Rise time (sec)	Over shoot %	Settling time (sec)	Steady state error (rad)
Roll	0.8	0.06	7	0
Pitch	0.9	0.3	4	0
Yaw	1	0	2	0.003
Altitude	2.7	0	3.5	0.05

Table 2: ANFIS controller hovering stabilization results.

## Results and Discussion

### Evaluation between the designed controllers

**First experiment:** Hovering stabilization desired  $[X; Y; Z, \psi]=[0, 0, 3, 0]$  the desired in this experiment is to reach the altitude of 3 m and taking into consideration that there is initial error in roll, pitch angles of 0.3 rad and yaw angle of 0.5 rad (Figure 10). In Figure 11, it can be seen that the output responses for roll obtained by modified PID controller is better than the output responses obtained by PID and ANFIS controller. There is a small oscillation and overshoot in PID and ANFIS controller. There is also a high oscillation for the ANFIS. The output response of pitch obtained by modified PID controller is better than the output responses obtained by other controllers. There is a small oscillation and overshoot in both PID and ANFIS controller. The output response of yaw obtained by modified PID controller is better than the output responses obtained by other controllers. There is an oscillation and overshoot in PID controller while steady state error with ANFIS controller. There is steady state error with ANFIS, modified PID and PID controllers for altitude channel.

The controller evaluation hovering stabilization results are shown in Table 3.

**Processor-in-loop experimental test:** The following test consists in the PIL (Processor-in-Loop) that goes beyond the PC platform. This step introduces some hardware features that permit to achieve situations that are more realistic where the control algorithm will run. In PIL, the target processor is a non-real time environment and

communication with external processors due to the use of specific functions installed in an integrated simulation environment installed in the host computer.

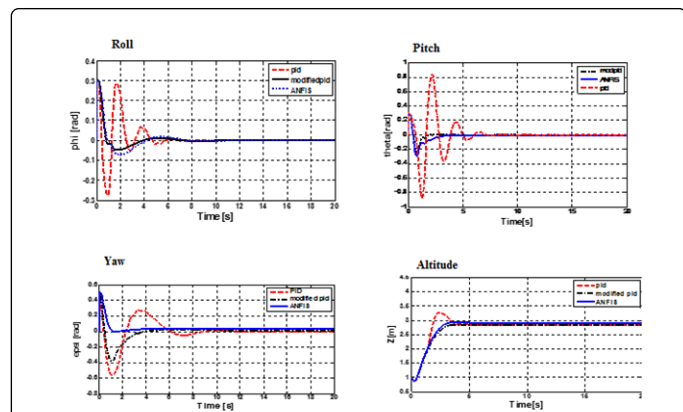


Figure 10: Output responses of the desired hovering stabilization.

Variables	Rise time(sec)			Over shoot%			Settling time(sec)		
	PID	Mod PID	Anfis	PID	Mod PID	ANFIS	PID	Mod PID	Anfis
Roll	0.6	0.9	0.9	0.28	0.03	0.04	8	4	4.1
Pitch	0.9	0.8	0.8	0.7	0.2	0.22	8	2	3
Yaw	1.2	0.9	0.99	0.59	0.4	0	8.5	5	2
Altitude	1.92	1.87	1.88	0	0	0	3.5	3.4	3.6

Table 3: Evaluation of designed controllers hovering stabilization.

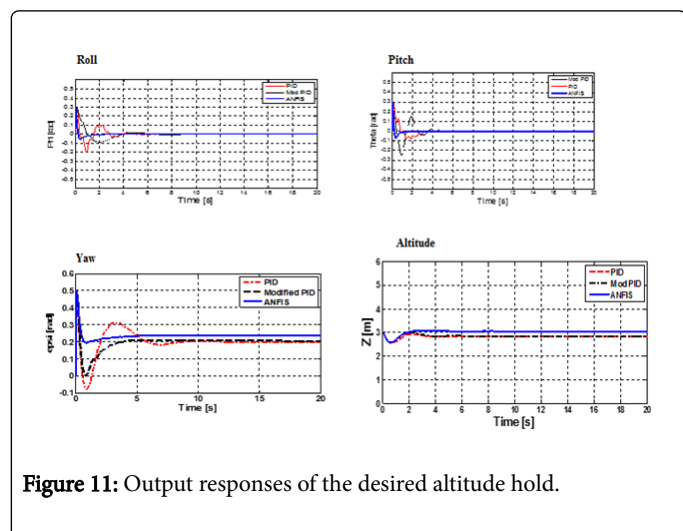


Figure 11: Output responses of the desired altitude hold.

In Figure 12, PIL block is used to interface and communicate with the processor. Develop, test and validate your embedded control code on the actual processor in the model. Embedded control algorithms are tested by executing it inside a circuit simulator. This can be done by executing actual code on real hardware as shown in the following Figure 12.

In Figure 13, PIL mode, the values calculated by the simulation tool are used as inputs for the built-in algorithm, instead of reading the actual sensors. Similarly, the output of the executed control algorithms on the processor is fed back into the simulation to drive the virtual environment. A data acquisition card is used to collect data from simulation model and send it to the embedded processor to resend it back to the simulation model. The data acquisition card is used also to overcome the problem of data transfer using serial connection in case of the two inputs. The input data to the controller in simulation is small float numbers but the serial block in MATLAB deal with unsigned integer numbers so a modified block is used as shown in following Figure 13.

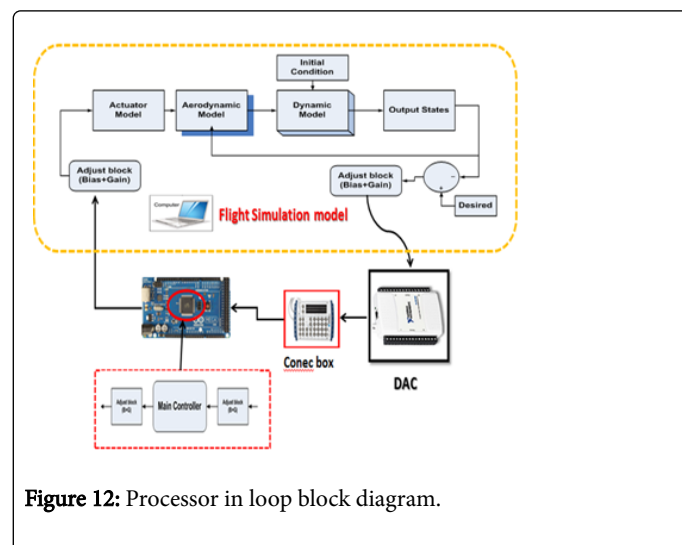


Figure 12: Processor in loop block diagram.



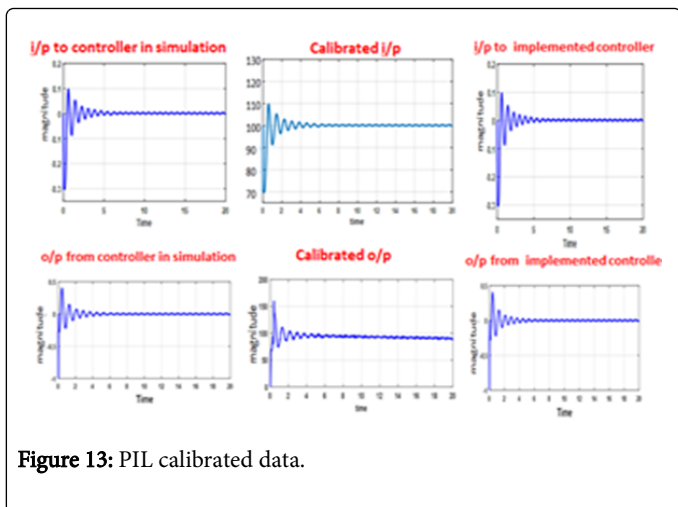


Figure 13: PIL calibrated data.

The modified block consists of two stages such that bias and gain block. The output from embedded controller is calibrated to overcome the problem of data transfer.

**PID-no processor in loop:** Desired  $[X; Y; Z, \psi] = [0, 0, 3, 0]$  the desired in this experiment is to reach the altitude of 3 m and taking into consideration that there is initial error in roll, pitch angles of 0.3 rad and yaw angle of 0.5 rad.

In Figure 14, the input is send to the controller in simulation and the embedded controller in the same time, but the loop closed by using the output from simulation.

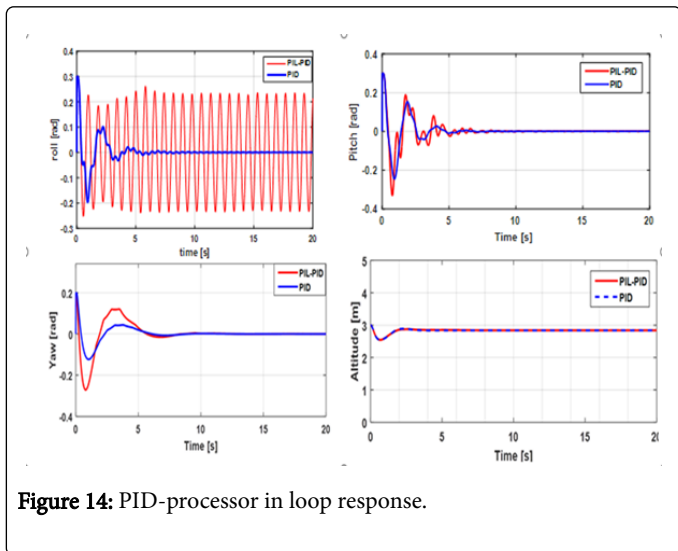


Figure 14: PID-processor in loop response.

The output response in roll has high oscillation due to high dynamic with the input to the embedded controller. The output response in pitch and yaw channels have changed due to data transfer but the output response in altitude has no change.

**MOD\_PID-processor in loop:** Desired  $[X; Y; Z, \psi] = [0, 0, 3, 0]$  the desired in this experiment is to reach the altitude of 3 m and taking into consideration that there is initial error in roll, pitch angles of 0.3 rad and yaw angle of 0.5 rad. In Figure 15 the input is send to the controller in simulation and the embedded controller in the same time, but the loop closed by using the output from embedded controller.

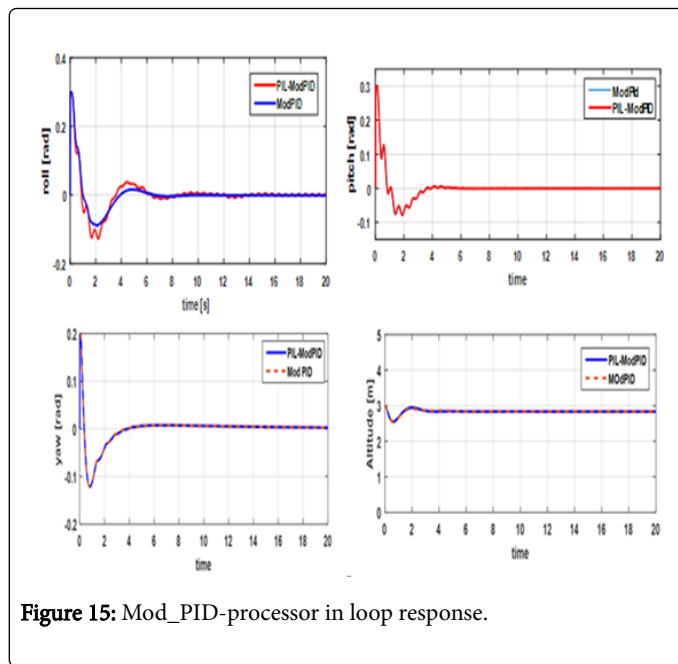


Figure 15: Mod\_PID-processor in loop response.

The output response in roll has a small change due to data transfer but the output responses in pitch, altitude and yaw have no change.

**ANFIS-processor in loop:** Desired  $[X; Y; Z, \psi] = [0, 0, 3, 0]$  the desired in this experiment is to reach the altitude of 3 m and taking into consideration that there is initial error in roll, pitch angles of 0.3 rad and yaw angle of 0.5 rad. In Figure 16 the input is send to the controller in simulation and the embedded controller in the same time, but the loop closed by using the output from embedded controller.

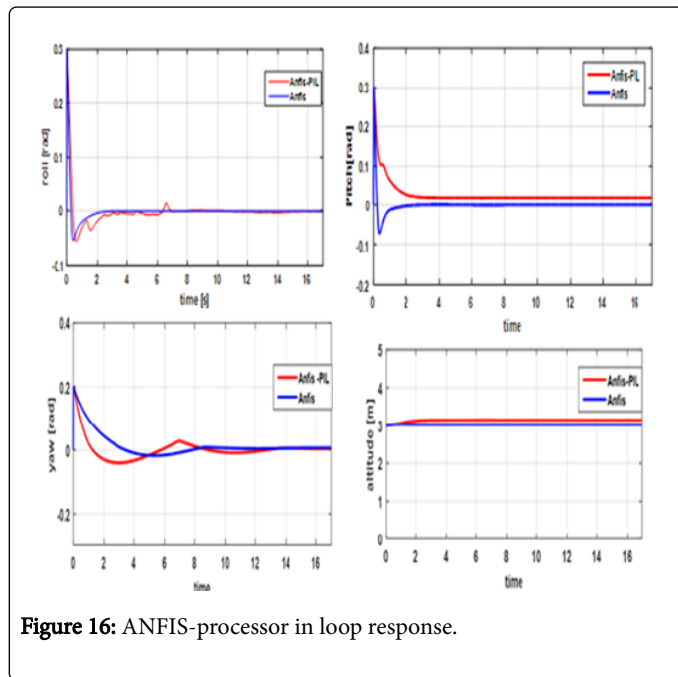


Figure 16: ANFIS-processor in loop response.

In Figure 17, the output responses in roll, pitch, altitude, and yaw have a change due to data transfer. This experiment shows that ANFIS has an execution time quadruple more than PID and modified PID controllers due to complex calculations.

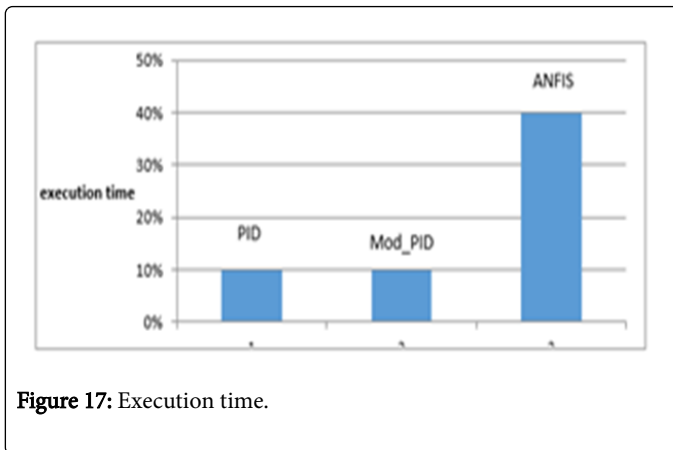


Figure 17: Execution time.

## Conclusion

The most important part of this paper was dedicated to finding a powerful and reliable control technique for quadrotor. The guidance and control system is designed using one of the control techniques such as modified PID, conventional PID controller and ANFIS controller. The designed controllers have the objective to circumvent some sources of uncertainty such as orientation variation. The evaluation results reveal that the designed modified PID controller has the best performance, more robust, higher stability and less control effort compared to the original and designed ANFIS controller. The modified PID has been selected for low control effort and less execution time via processor in loop experiment. This gives the green light for the next step for the implementation of modified PID

controller on embedded board. Towards the verification of the designed controller, evaluation of the intended system was tested on real environment, and the real flight response was obtained and evaluated.

## References

1. Castillo-Effen M, Castillo C, Moreno W, Valavanis KP (2007) Control fundamentals of small/miniature helicopters -A survey. In: *Advances in Unmanned Aerial Vehicles* 33: 73-118.
2. Tayebi A, McGillvray S (2006) Attitude stabilization of a VTOL quadrotor aircraft. *IEEE Trans Control Syst Technol* 14: 562-571.
3. Martinez VM (2007) Modelling of the flight dynamics of a quadrotor helicopter. Msc Thesis, Cranfield University, Bedfordshire, UK.
4. Xu R, Ozguner U (2006) Sliding mode control of a quadrotor helicopter. *45th IEEE Conference on Decision and Control* pp: 4957-4962.
5. Bouabdallah S, Siegwart R (2005) Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. *IEEE International Conference on Robotics and Automation* pp: 2247-2252.
6. Dingjie W, Liang Z, Jie W, Guanghua L (2014) Research on the real-time information fusion algorithm for GNSS/MIMU integrated navigation system. *IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)* pp: 2028-2033.
7. Bouabdallah S, Murrieri P, Siegwart R (2004) Design and control of an indoor micro quadrotor. *IEEE International Conference on Robotics and Automation* pp: 4393-4398.
8. Ogata K (2009) *Modern control engineering* (5th edn). Lugar: Upper Saddle River, Prentice Hall, New Jersey, USA.
9. Hashemi A, Fitzpatrick B, Wang LY, Yin G (2014) Robust noise attenuation under stochastic noises and worst-case unmodelled dynamics. *Int J of Sys Sci* 45: 1563-1578.