opción

Revista de Antropología, Ciencias de la Comunicación y de la Información, Filosofía, Lingüística y Semiótica, Problemas del Desarrollo, la Ciencia y la Tecnología

Año 35, 2019, Especial N° 19

Revista de Ciencias Humanas y Sociales

ISSN 1012-1587/ ISSNe: 2477-9385

Depósito Legal pp 198402ZU45

Universidad del Zulia

Facultad Experimental de Ciencias

Departamento de Ciencias Humanas

Maracaibo - Venezuela

Opcion, Año 35, Nº Especial 19 (2019):*2899-2921*
ISSN 1012-1587/ISSNe: 2477-9385

Estimation and Control of an UAV using Kalman Filter and Relay Feedback Autotuning Method

[1]Laith S. Ismail, [2]Ciprian Lupu

[1] Automatic Control and Computers Science Politehnica University of Bucharest, Computer Techs. Eng. Dep. Al Turath University Baghdad, Iraq; laith_ismail@turath.edu.iq
[2] Automatic Control and Computers Science Politehnica University of Bucharest Bucharest, Romania; ciprian.lupu@acse.pub.ro

Abstract
An unmanned aerial vehicle (UAV) also known as Unmanned Aerial Systems (UAS) or Drone. UAV can be a remote-controlled aircraft or pre-programmed route. UAV are gaining popularity and demand in the commercial sector and for personal, hobbyist use. There is an increasing demand for UAV in areas such as thermal scanning, delivery, and mapping in areas, which are time-consuming, difficult, or dangerous for a conventional approach. The design discussed in this paper is basing on the development of UAV quadrotor helicopter (Quad Copter), software, modeling, estimation, and control system. The estimation depends on Kalman filter and the control part depend on Relay feedback autotuning PID control method. The implementation is divided into two parts, simulation by using (Matlab) and real-time Implementation by using (LabVIEW and Arduino). The solution in this paper is used low-cost devices and tools such as Arduino and its use in building an educational platform and linking it with an industrial program such as the LabVIEW.

Keywords: quadcopter, UAV, Relay feedback method, PID control, Kalman filter, Real time implementation.

Estimación y control de un UAV utilizando el método de ajuste automático de retroalimentación y filtro de Kalman

Resumen
Un vehículo aéreo no tripulado (UAV) también conocido como Sistemas aéreos no tripulados (UAS) o Drone. UAV puede ser un avión a control remoto o una ruta preprogramada. Los UAV están ganando popularidad y demanda en el sector comercial y para uso personal y de aficionados. Hay una creciente demanda de UAV en áreas como el escaneo térmico, la entrega y el mapeo en áreas que requieren mucho tiempo, son difíciles o peligrosas para un enfoque convencional. El diseño discutido en este documento se basa en el desarrollo del helicóptero quadrotor UAV (Quad Copter), software, modelado, estimación y sistema de control. La estimación depende del filtro de Kalman y la parte de control depende del método de control PID de autoajuste de retroalimentación de relé. La implementación se divide en dos partes, la simulación usando (Matlab) y la implementación en tiempo real usando (LabVIEW y Arduino). La solución en este documento es utilizar dispositivos y herramientas de bajo costo como Arduino y su uso en la construcción de una plataforma educativa y vincularla con un programa industrial como LabVIEW.

Palabras clave: quadcopter, UAV, método de retroalimentación de relé, control PID, filtro de Kalman, implementación en tiempo real.

Introduction

Unmanned aerial vehicle is one of the most complex flying machines because of Fluency of movement and maneuverability to execute a number of tasks. Classical helicopters are generally balanced with the base rotor and a tail rotor. In this paper is presented a kind of Unmanned aerial vehicle known as a quadcopter.[1]

The lift force is displacing in the x and y-axes, because of either a roll or a pitch Respectively, Leading to a horizontal force component that will direct the quadcopter. UAV representation is obtaining by representing the quadcopter as a solid body converting in Three-dimensional to three moments with one force as shown in fig. 1. The general coordinates of quadcopter are:

$$q=(x,y,z,\Psi,\Theta,\varphi)$$

Where (x, y, z) indicates to the position of the center of the mass of the vehicle in relation to the frame base and (Θ,φ,Ψ) represent the three angles yaw, pitch and roll (Euler angles) and indicates the orientation of the quadcopter.

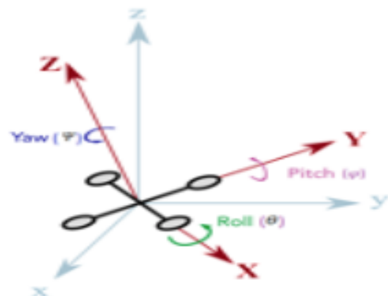


Fig. 1. Quadcopter General Inertial Frame Coordinates.

II. THE MECHANISM OF OPERATION OF QUADCOPTER

The quadcopter has four equal motors and propellers generating four thrust forces and there are two possible configurations for quadcopter: "+" or "×" motors 1 and 3 rotate CW, 2 and 4 rotate CCW and by depending on Third Newton's Law Required to compensate the action and reaction effect. [2] [3]

All thrusts have the same direction because propellers one and three have opposite pitch (w.r.t.) two and four. The forces and rotation speed can analysis to include:

1. ω_{1-4} : the rotation speeds from motors
2. T_{1-4} : forces from motors
3. $T_i \propto \omega_i^2$: depend on the propeller (air density, shape, ...)
4. m : the mass of the quadcopter
5. Mg : the weight of the quadcopter
6. M_{1-4} : moments from the forces
7. $M_i = L \times T_i$

They have two reference systems as shown in fig. 2:

1. The inertial reference system, (the Earth-fixed frame) (x_E, y_E, z_E)

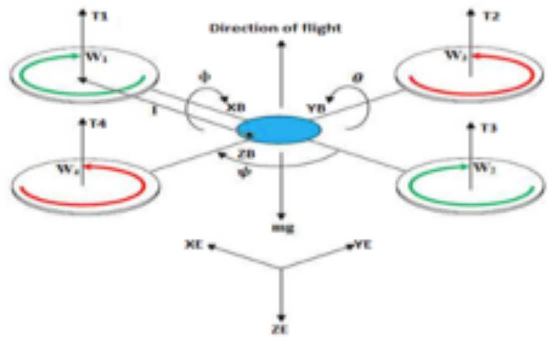


Fig. 2. Dynamic of a Quadrotor

2. The quadcopter reference system, (the Body-fixed frame) (x_B, y_B, z_B)

We connect Euler Angles (φ, θ, ψ) to set the conversion between the two systems:

Roll (φ): will be angle of rotation o n the axis $X_B||X_E$

Pitch(θ): will be angle of rotation on the axis $Y_B||Y_E$

Yaw (ψ): will be angle of rotation on the axis $Z_B||Z_E$

The derivative of (φ, θ, ψ) are rotation speeds and the angular $(\dot{\varphi}, \dot{\theta}, \dot{\psi})$ of the quadcopter and will be $(\dot{\varphi})$ Roll rate, $(\dot{\theta})$ Pitch rate and $(\dot{\psi})$ Yaw rate.

Control System Design

Quadrotor control is a principally complex and motivating problem. The most interesting issue is with 6 degrees of freedom and 4 independent inputs, it will be under actuated and the resulting dynamics will be extremely nonlinear. It is different from normal vehicles, UAVs have a resistance to Movement obstruction, Consequently, UAV should supply them own damping to reduce moving and hold themselves in stability region. For these reasons, the problem of stability and control of the quadcopter is attractive and interesting.

Control strategy

To achieve the stability of the quadcopter system, the exemplary plan is to use 3 PID control loops that measure the current roll, pitch, and yaw; given by (θ,φ,ψ) and the alteration in the particular quantities ($\dot{\theta}$,$\dot{\varphi}$,$\dot{\psi}$) relate to some desired positions. The control for the change in posi-

tions and angles is by the users from a radio, computer, remote control, or pre-programmed plans [4]. Tuning the gains is a very hard mission, for an unstable system. Although the parameters of PID can be adjusted manually offline but it takes a lot of time to tune the gains for the quadcopter and in the same time is an inaccuracy. In our application, the PID controller is tuned using classical tuning method (Ziegler-Nichols) [5] in simulation case (Matlab) and modern autotuning method (Relay Feedback) in real time case.
In this strategy as shown in fig.4. Now it have these parameters:
The measures:

- The actual angular velocities to the 3-axis $(\dot{\phi}_M, \dot{\theta}_M, \dot{\psi}_M)$
- They measure from sensors (MPU-6050)

The set points:

- The desired angular velocities to the 3-axis $(\dot{\phi}_T, \dot{\theta}_T, \dot{\psi}_T)$
- From computer or remote control

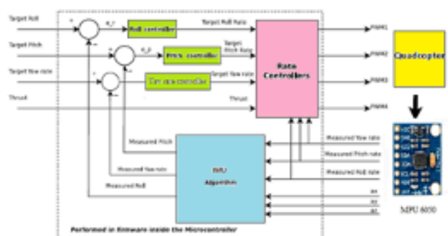


Fig. 3. the overall control loops

We can do control loop by using this algorithm:

While true do

On ΔT *timer tick* ;

$(\dot{\phi}_T, \dot{\theta}_T, \dot{\psi}_T, F) = simple\_remote\_control()$;

$(\dot{\phi}_M, \dot{\theta}_M, \dot{\psi}_M, F) = simple\_gyro()$;

$e_{\dot{\phi}} := \dot{\phi}_T - \dot{\phi}_M \quad e_{\dot{\theta}} := \dot{\theta}_T - \dot{\theta}_M \quad e_{\dot{\psi}} := \dot{\psi}_T - \dot{\psi}_M$

$C_{\dot{\phi}} := Roll\_rate\_control(e_{\dot{\phi}})$;

$C_{\dot{\theta}} := Pitch\_rate\_control(e_{\dot{\theta}})$;

$C_{\dot{\psi}} := Yaw\_rate\_control(e_{\dot{\psi}})$;

$(pwm_1, pwm_2, pwm_3, pwm_4)^T = K^{-1}(C_{\dot{\phi}T}, C_{\dot{\theta}T}, C_{\dot{\psi}T}, F)^T$

END

Where C is in the controllers. It used here PID controller:

$$C := rate\ controller(e);$$

Will be:

$$C(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

In a discrete world (at k^{th} sampling instant):

$$C(k) = K_p e(k) + K_i \int_0^k e(j) \Delta T + K_d \frac{e(k) - e(k-1)}{\Delta T}$$

IV. SIMULATION IMPLEMENTATION

A. Simulation Implementation by using Matlab/Simulink

In this section will try to implement what we did in the previous sections in simulation model. Matlab/Simulink is chose to do that. Two-control loop (rate control (inner) and attitude control (outer)) use here for the first one is used PID controller and for the second one is used PI controller. By using a classical tuning method Ziegler-Nichols to tuning parameters of PID. As shown in fig.5.

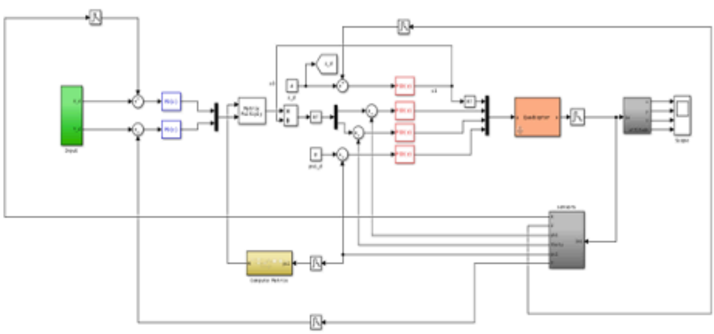
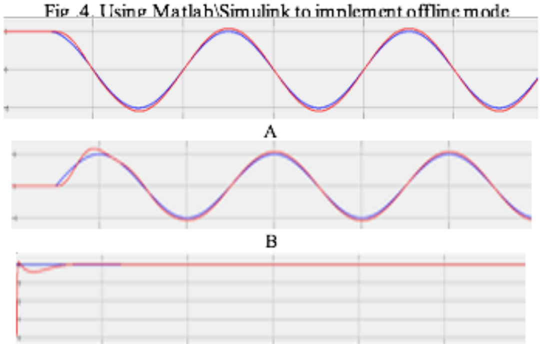


Fig .4. Using Matlab\Simulink to implement offline mode



A

B

C

Fig .5. the x, y and z components (Red), respectively (a, b and c), while following the targets (Blue).

and fig.6. [10]

B. Simulation Implementation by Using LabVIEW

Before start the implementation on the platform. Simulation are creating simulating the movement of the plane in the LabVIEW and linking this simulation to the remote control. The implementation of the process is in real time and simulation in the same time by using this remote to understanding the process before the online implementation on the platform.

This 3D simulation is depending on the mechanism of operation of Quadcopter that explained in previous chapters. This simulation was built using 3D picture control tools in LabVIEW program.[11] The remote control was used the joystick from Microsoft, it is name (Controller Microsoft Xbox One wireless), as shown in figure (6).

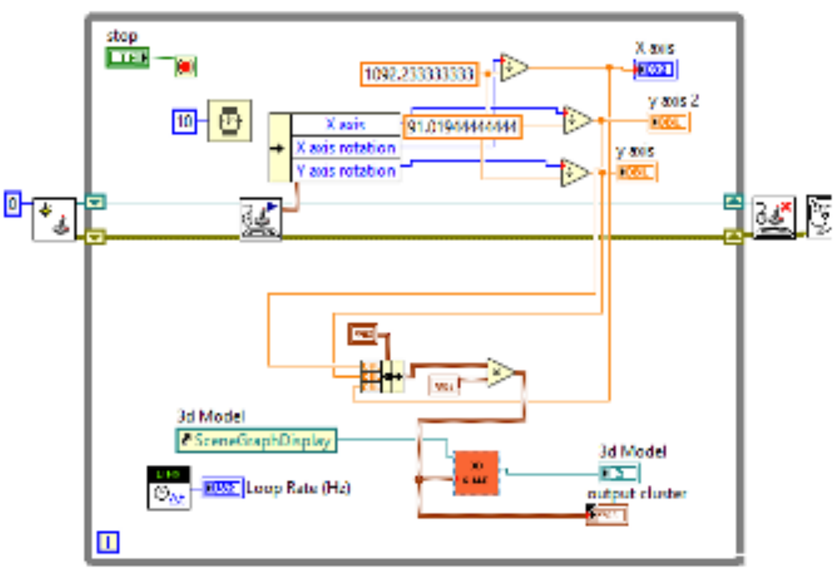


Fig .7. Internal Program of 3D simulation of quadcopter

V. Real Time Implementation

This paper divided into theoretical and practical work; the particular needs a suitable platform to achieve the results. This platform is simulation the quadcopter in real time. All parts of the platform designed locally. This contains from hardware, software. of this prototype. Will be using in this paper the LabVIEW program and platform beside Arduino IDE because LabVIEW is one of the best industrial programs at this moment and there are a lot of tools we can use it in our paper. The most important challenge is how can connect LabVIEW with Arduino. There way is simple but not advance it is by using LabVIEW Interface for Arduino toolkit. This way is tried but we saw there are some parts not available with this toolkit for example Gyroscope MPU6050 sensor it was not support with this toolkit, we tried to build it by using the Virtual Instrument Software Architecture (VISA) Technique in LabVIEW to build I2C signals but the result was not good. Therefore, another way is implemented, by choosing work with LINX toolkit with VISA to make interface between LabVIEW with Arduino mega. The LINX toolkit by MakerHub group makes it easy to interface in real time and working with common embedded platforms such as Arduino, and NI myRIO, chipKIT as well as most of sensors including accelerometers, gyroscopes, temperature sensors, and most common sensors. With features of NI LabVIEW software and LINX toolkit, it can acquire or control data from Arduino. When the information arrives from Microcontrollers, that can use it and analyze by using a lot of built-in LabVIEW libraries, develop methods to work on hardware, software, and present your existing on a GUI. [13]

LINX supplies tools for popular embedded platforms that act as an I/O devices and interfaces with LabVIEW platform through a wireless, serial, Ethernet connection or USB. This helps the developers to works with moving the information from microcontrollers devices to LabVIEW without edit the communication, synchronization, or even a C code.

Estimation of Roll and Pitch

To facilitate the readings data from the IMU sensor, the data should be scaled to get the gyroscope and accelerometer measurements in rad/s and m/s respectively. Moreover, gyroscope and accelerometer bias must be estimated and removed them. By using, the basis accelerometer type equations and resolving for φ and θ gives:

$$\varphi_{acc} = tan^{-1}\frac{a_x}{\sqrt{a_y^2 + a_z^2}} \quad (2)$$

$$\theta_{acc} = tan^{-1}\frac{a_y}{\sqrt{a_x^2 + a_z^2}} \quad (3)$$

The pitch and roll can be estimated from the gyroscope data as:

$$\varphi_{gyro} = \int_0^t p(\tau)d\tau \quad (4)$$

$$\theta_{gyro} = \int_0^t q(\tau)d\tau \quad (5)$$

Where p, q and r are the bodies fixed angular-velocity vector .Or discretely, using Euler's method, as

$$\varphi_{gyro}[n] = \varphi_{gyro}[n-1] + h.p \quad (6)$$

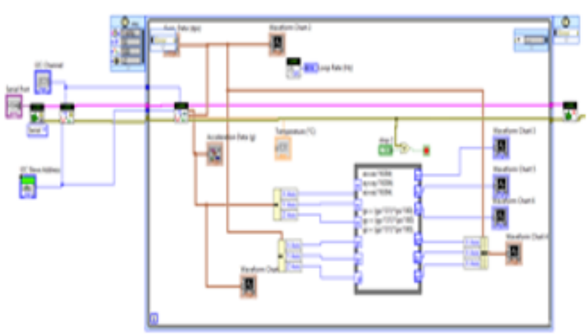


Fig .10. Gyroscope implementation in LabVIEW

$$\theta_{gyro}[n] = \theta_{gyro}[n-1] + h.q \quad (7)$$

B. Kalman Filter

The rate gyroscope measures velocity in each axis (x, y, z) and therefore to get position it need to integrate this measurement. The trouble is that gyro measurements drift over time and have a bias. The accelerometer can also measure angle using the arctanfunction but it is very sensitive to high frequency vibrations and so it too has difficulties, but it does not drift. Therefore a system is needed which combines the two reading in what we call sensor fusion. The precise angle is calculated the precise angle by using a technique called a Kalman filter [7].

We can now combine the gyroscope and accelerometer measurements, it can be obtained from the Extended Kalman Filter as presented in Section above and applied it in LabVIEW. The Kalman filter needs the output Jacobian matrices and the system spreading functions. From the kinetic theory:

$$\dot{\varphi} = p + q\, sin(\varphi)\tan(\theta) + r\cos(\varphi)\tan(\theta) + \zeta_\varphi \qquad (8)$$

$$\dot{\theta} = q\, cos(\varphi) - r\sin(\varphi) + \zeta_\theta \qquad (9)$$

By using the simplified calculation model equations and by setting$x = (\varphi, \theta)^T$, $y = (ax.ay, az)^t$, $u = (p, q, r, u, v, w)^T$ *and* $\zeta = (\zeta_\varphi; \zeta_\theta)^T$ produce

$$\dot{x} = f(x, u) + \zeta \qquad (10)$$

$$y = h(x, u) + \eta \qquad (11)$$

When

$$f(x, u) = \begin{bmatrix} p + q\, sin(\varphi)\tan(\theta) + r\cos(\varphi)\tan(\theta) \\ q\, cos(\varphi) - r\sin(\varphi) \end{bmatrix} \qquad (12)$$

$$h(x, u) = \begin{bmatrix} g\, sin(\theta) \\ -g\, cos(\theta)\sin(\varphi) \\ -g\, cos(\theta)\cos(\varphi) \end{bmatrix} \qquad v \quad (13)$$

Which output the Jacobians

$$\frac{\partial f}{\partial x} = \begin{pmatrix} p\cos(\varphi)\tan(\theta) - r\, sin\tan(\theta) & \frac{qsin(\varphi) - r\, cos(\varphi)}{cos^2(\theta)} \\ -q\, sin(\varphi) - r\, cos(\varphi) & 0 \end{pmatrix}$$

$$\frac{\partial h}{\partial x} = \begin{pmatrix} 0 & g\cos(\theta) \\ -g\, cos(\varphi)\cos(\theta) & g\sin(\varphi)\sin(\theta) \\ g\sin(\varphi)cos\,(\theta) & g\, cos(\varphi)sin\,(\theta) \end{pmatrix} \qquad (14)$$

Moreover, the Kalman filter needs that set the matrices x_0, R, P and Q. it can start from the surface, let's say that put $x_0 = 0$ and P = I_{3x3}. Q and R are tuning matrices. For implementation, this

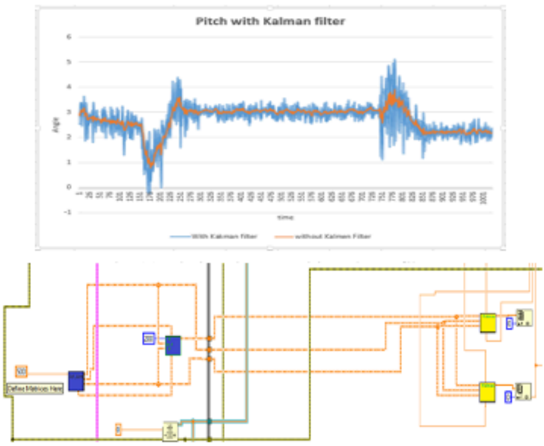


Fig .12. Kalman filter implementation in LabVIEW

algorithm is used LabVIEW to build it and it is not depend on any toolkit to build it. [8]

C. PID autotuning algorithm based on Relay Feedback Method

In the last years, the relay feedback method has set a new technique in the automatic tuning of PID controllers and in the configuration of other developed adaptive controller. In this paper is used relay feedback method as autotuner parameters of PID controller. This method is depended on the estimate of the critical point on the response of operating frequency by applying relay signal oscillations. Now can obtain the important process information and a relay feedback experiment, it can be generating a constant cycling of the controlled changing. Ultimate period (Tu) and Ultimate gain (Ku) can be obtained from the experiment signal directly. In relay feedback auto-tuning, the relay is connected in the feedback loop as shown in figure (13) to obtain limit period oscillation. The system is usually analyzed using describing function [9].

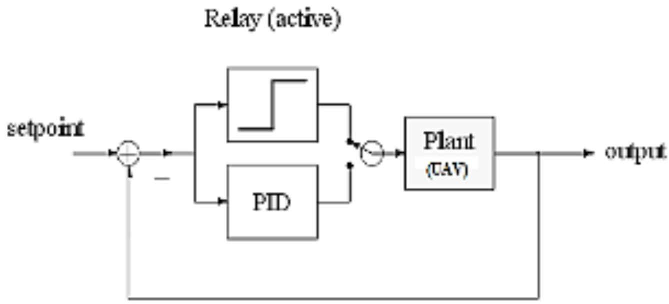


Fig. 13. Relay feedback method

Because a relay is a nonlinear element, that should to find some linear representation by using some mathematical tools. The best approach by using "describing function." The relay is replaced by a dependent gain N_L (a), and (a) is the size of relay input. [12]

$$G(i\omega_u) = \frac{-1}{N_L\ (a)} = \frac{\pi\ a}{4h} \qquad (15)$$

The imaginary part of $G\,(i\omega_u)$ is zero. Thus, the frequency is the ultimate frequency of the process and the ultimate gain is$1/G\,(i\omega_u)$. That is mean:

$$\omega_u = \frac{2\pi}{T_u} \qquad (16)$$

$$K_u = \frac{4h}{\pi\ a} \qquad (17)$$

$$G(i\omega_u) = \frac{-1}{N_L\ (a)} = \frac{\pi\ a}{4h} \qquad (15)$$

The imaginary part of $G\,(i\omega_u)$ is zero. Thus, the frequency is the ultimate frequency of the process and the ultimate gain is$1/G\,(i\omega_u)$. That is mean:

$$\omega_u = \frac{2\pi}{T_u} \qquad (16)$$

$$K_u = \frac{4h}{\pi\ a} \qquad (17)$$

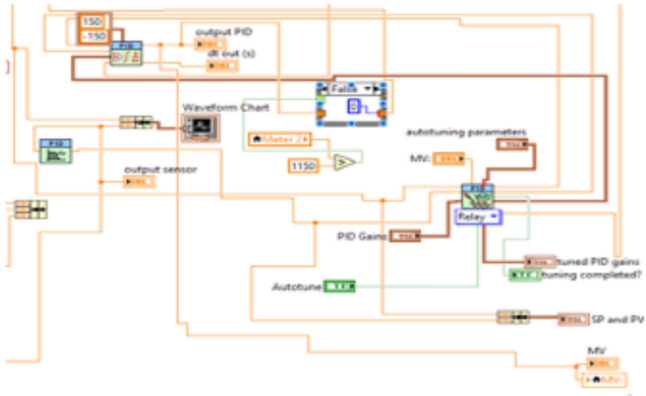


Fig .14. Tuning PID control by using Relay feedback method in LabVIEW [6]

VI. Experments and results

The one degree of freedom is allowing us to test the control for Pitch and Roll Euler angles without Yaw because it is impossible to test it with one degree of freedom. The experimental procedure involves testing the actual pitch angle and control it by using close loop theories. By using Relay feedback method for autotuning PID controller, from user interface can choose the relay's parameters, apply it to the plant, and turn on the autotuning. Relay signal is sent to the platform and the relay's parameters are the amplitude of relay (h=50), the relay cycles = 3, and the amplitude of oscillation (a = 31.88).

From eq. (17) the ultimate gain is $Ku = (4\times50/\pi\times31.88) = 1.996$. The ultimate period is ($Tu = 263.99$). So the tuned PID's parameters are $Kp = 1.1976$, $Ti = 0.001894$ and $Td = 0.000455$. The response of the system in close loop case is in figures (16) and (17).

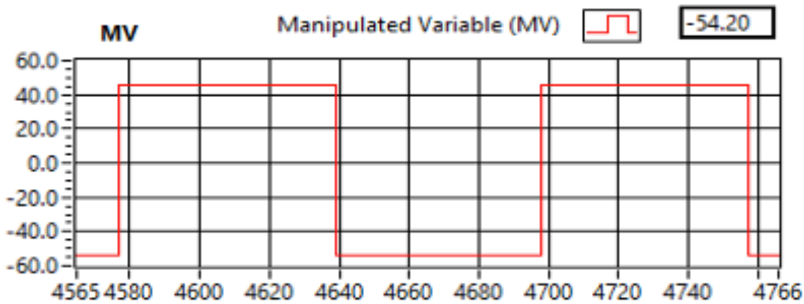


Fig. 15. Relay signal

Conclusion

The quadcopter is complex systems which design requires knowledge of the theoretical aspects. Because this paper is educational paper, the basic principles was used to design the practical part and applied these aspects with simulation in the Matlab and the three-dimensional simulation in the LabVIEW.

The first challenge was connecting LabVIEW with Arduino because the common way to connect it by using LabVIEW Interface for Arduino toolkit but this way it was not good with some parts. This problem was overcame by using new technique combine between LINX toolkit and VISA package. After the connection and hardware implementations was finished, the desired signals (accelerometer and gyroscope) were estimated to obtain the Euler angles and the main problem was the noise and drift the signals, this problem is solved by using an intelligent technique call as (Kalman filter), The Kalman filter chosen is nonlinear version (Extended Kalman filter). Then the control system theory is designed to work in real time. PID controller is used to control the Euler angles (pitch and roll). The PID's parameters are tuned by using Autotuning method called (The Relay Feedback). Experiment results in real time is shown the robustness, stability, and error boundedness of the PID controller with Relay Feedback method.

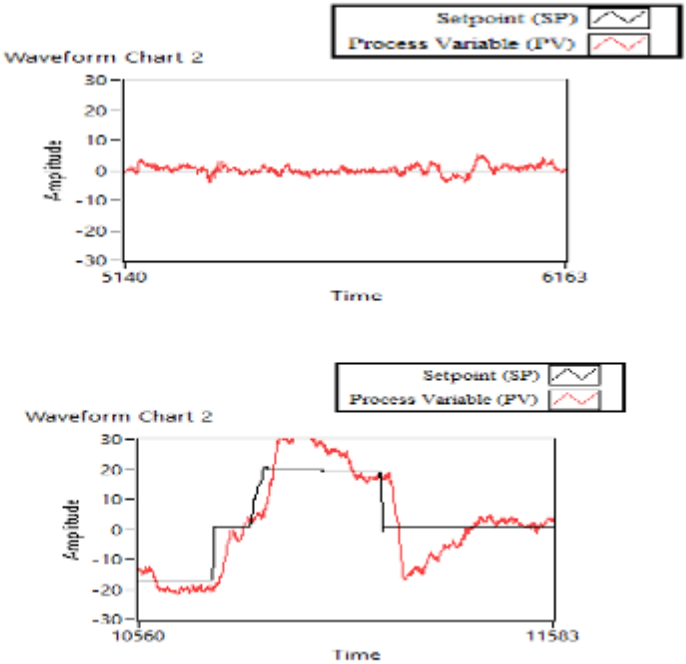


Fig.17. the response of System with tuned coefficients (Pitch angle)

One of the most important issues of this paper is used low-cost solutions and tools such as Arduino and its use in building an educational platform and linking it with an industrial program such as the LabVIEW.

References

[1] "ICAO's circular 328 AN/190: Unmanned aircraft systems". ICAO. Retrieved 3 February 2016.
[2] Luukkonen T.: Modelling and Control of a Quadcopter.
[3] Benić, Z.; Piljek, P. and Kotarski, D.: Mathematical modelling of unmanned aerial vehicles with four rotors.
[4] I. Diemen, A. Arisoy, and H. Temeltas, "Attitude control of a quad-rotor," in Recent Advances in Space Technologies, 2009. RAST'09.

4th International Conference on, pp. 722–727, 2009.
[5] Katsuhiko Ogata, “Modern Control Engineering”.5th pp 567-648, 2008.
[6] National Instruments,” LabVIEW PID and Fuzzy Logic Toolkit User Manual”, June 2009.
[7] R. W. Beard and T. W. McLain. Small unmanned Aircraft Theory and Practice. Princeton, 2012.
[8] T. I. Fossen. Handbook of Marine Craft Hydrodynamics and Motion Control. Wiley, 2011.
[9] S. Levy, S. Korotkin, K, Hadad. A. Ellenbogen, M. Arad, Y. Kadmon, “PID autotuning using Relay feedback”27th Convention of Electrical and Electronics Engineers in Israel, IEEE, 2012.
[10] P. I. Corke, Robotics, Vision&Control: Fundamental Algorithms in Matlab. Springer, 2011.
[11] LabVIEW Software. http://www.ni.com/ro-ro/shop/labview.html Accessed: 2019-03-13.
[12] Mary Jermila M, Anju Iqubal, Soumya Raj. “Pid tuning using Relay Feedback” Vol. 2 Issue 4, April – 2013.
[13] LINX Software. <https://www.labviewmakerhub.com>. Accessed: 2018-2-4.

UNIVERSIDAD
DEL ZULIA

opción
Revista de Ciencias Humanas y Sociales

Año 35, Especial N° 19, 2019

Esta revista fue editada en formato digital por el personal de la Oficina de Publicaciones Científicas de la Facultad Experimental de Ciencias, Universidad del Zulia.
Maracaibo - Venezuela

www.luz.edu.ve

www.serbi.luz.edu.ve

produccioncientifica.luz.edu.ve