IoT Based UAV Platform for Far Distance control system

1st Dr. Amer Farhan Sheet, 2nd Fahad Yassir Al Bazzaz[2], 3rd Dr. Omar I. Dallal Bashi
4th Mustafa Zuhaer Nayef Al-Dabagh
{ dr.amerfarhan@ntu.edu.iq[1], fahad.albazzazapple@gmail.com[2], omardallalbashi@ntu.edu.iq, mustafa.nayef@knu.edu.iq[4] }

Northern Technical University, Mosul, Iraq[1,3], State Company of North Distribution Electricity, Ministry of Electricity, Iraq[2], Department of Computer Science, College of Science, Knowledge University University Park, Kirkuk Road, Erbil EBL 5500, Kurdistan Region – Iraq[4.]

Abstract. Unmanned Aerial Vehicles (UAV) are becoming more common in our modern world. UAVs are mostly associated with war due to the coverage of their use in the recent wars in Iraq and Syria, but have the ability to do much more. UAVs are helpful tools in assessing damage after a disaster, keeping rescuers safe while they help those in need. UAVs are useful tools in monitoring crops to ensure the maximum yield is realized. The use of UAVs is also being used for monitoring remote land areas that are difficult to reach by foot. However, uses of UAVs are endless. This paper goal, is designing a system that uses internet of things (IoT) to make the control for UAV (drone) from an infant distance by simulating the signals coming from RF receiver and remote control, these signals will be provided from Raspberry Pi3 and Arduino Uno devices to apply the signals of Roll, Pitch, Yaw, and Throttle to take control of UAV, the system uses an interaction between raspberry pi and flight controller to overcome the complexity of stability and PID control calculations. The control is designed using python 2.7 GUI (Graphical User Interface) to take control for flight, also the design is containing the capability of tracking objects based on the BGR color moment calculations.

Keywords: Internet of Things, Drone, PID (Proportional-Integral-Derivative).

1 Introduction

The Internet of Things is already a well-known term nowadays and it is attractive increasingly important as various types of systems and sensors, it is advanced to assist people in their daily lives. Connected devices number continues to increase worldwide. And also the variety and the applications in the real world are enormous, making it an attractive manufacture to action on **[1]**.

Many associations and firms are active and industrious on modernistic devices and new resolutions to transact with the increase of the joined devices, which is a rapidly rising industry. It has the ability to overturn any singular milieu such as transportations, seeding, industrializations, intelligent houses even entire metropolis. Associations and firms are seeking to make new devices for any conceivable scenario but they are also meliorating security as well as communication protocols. because of costs continue to decrease and claim continues to increase, It is rated that about 0 billion devices were connected in 2015, 20 to 30 bilion devices will be connected by 2020 **[2]**.

IMDC-SDSP 2020, June 28-30, Cyberspace

DOI 10.4108/eai.28-6-2020.2297949

With contemporary progressions in technology, it has become more potential to address a larger number of problems concerning accessibility and likewise to monitor systems more thoroughly and to interact with them. The capabilities of the system is established on the use of unmanned surface vehicle (USV) which help with nautical monitoring tasks by using multi-drone swarms **[3]** for an expanded spatial solutions. The system is able to discover things such as oil spills **[4]**, or it may be used for the general monitoring of environmental conditions.

The Internet of Things can assist to settle many problems in various domains: including intelligent metropolis; IoT implementations are concerning with noise, parking issues, traffic, and illumination monitoring **[5]**; contingency system for earthquakes **[6]**; and thoroughness agriculture applications in culture operation optimization **[7]**. The Internet of Things (IoT) is used to convey information from the sensors and to the actuators.

The application of UAVs can be separated into military and urbanized models. The antecedent is intended for non-governmental and governmental purposes; for example, the employment of UAVs in salvage operations and in recovery from large-scale catastrophe events, like as "the great East Japan earthquake" **[9]**, "the natural disasters of Indonesia" **[10]**, and "the earthquake of Nepal" **[11]**.

Nevertheless, in the very near future, UAV will not only be applied for generic preservations and catastrophe assist operations **[12, 13]** but likewise into many other civilians, governmental and merchant services. Some kindly examples of these are reconnoitring and surveillance **[14]**, native land security **[15, 16]**, generic safety **[17]**, ecological monitoring **[18]**, forest combustion monitoring **[19]**, safeness and border oversight **[12]**, cultivation [13], "Internet delivery" **[20, 21]**, architectural observation **[22]**, "goods transportations" **[23, 24]** like "Amazon Prime Air" **[25]**; which is prepared to safely transfer packages to customers during 30 minutes by the use of small UAV. With their countless implementations, Drones will shortly be an influential part of our life and will comprise an indispensable technology that is identical to the smartphones of todays.

The Contribution of this paper is designing a system that uses an Internet of Things (IoT) to make the control for Unmanned Aerial Vehicle (drone) from an infant distance by simulating the signals coming from RF receiver and remote control, these signals will be provided from raspberry pi device to apply the signals of Roll, Pitch, Yaw, and Throttle to take control of UAV, the system uses an interaction between raspberry pi and flight controller to overcome the complexity of stability and PID control calculations.

The rest of this paper is organized as follows. The first section about general concepts of IoT and UAV. The second section about system hardware components that been used in our work. The third section introduced the system simulation and PID parameter are showed.

2 Modeling of UVA dynamic and Kinematics

We will commence deriving UAV dynamics by introducing the two frames in which will operate. The inertial frame is defined by the ground, with gravity pointing in the negative z-direction. The body frame is defined by the orientation of the UAV, with the rotor axes pointing in the positive Z-direction and the arms pointing in the X and Y directions. Figure 1 shows the UAV body frame and inertial frame.

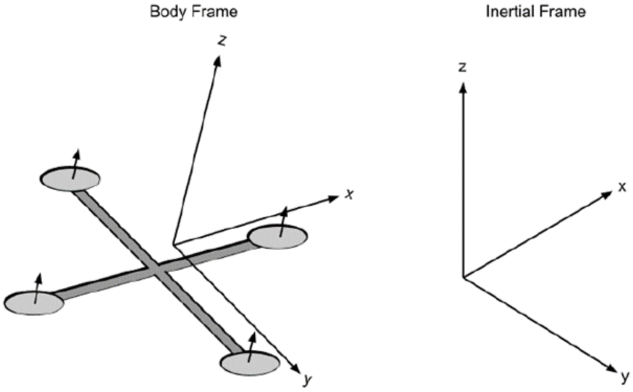

Fig. 1. UAV Body frame and Inertial frame.

2.1 Kinematics

Let us formalize the kinematics in the body and inertial frames before delving into the physics of UAV motion. Position and velocity of UAV defined in the inertial frame as $position = (x, y, z)^T$and $city = (\dot{x}, \dot{y}, \dot{z})^T$. Similarly, the Yaw, Roll, and Pitch angles in the body frame was defined as $\theta = (\phi, \theta, \psi)^T$, with identical angular velocity as $\dot{\theta} = \left(\dot{\phi}, \dot{\theta}, \dot{\psi}\right)^T$. However, with the consideration that the angular velocity vector $\omega \neq \dot{\theta}$. The angular velocity is a vector pointed along the rotational axis, while $\dot{\theta}$ is the time derivative of Roll, Yaw and Pitch. "To convert these angular velocities into angular velocity vector", we can utilize the following relation:

$$\omega = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & C_\theta S_\phi \\ 0 & -S_\phi & C_\theta C_\phi \end{bmatrix} \dot{\theta} \tag{1}$$

Where ω is the angular velocity vector in the body frame.

The body frame can related with inertial frame by a rotation matrix R that goes from the body frame to the inertial frame. This matrix is determined by utilizing the ZYZ Euler angle conventions and successively "undoing" the Pitch, Roll and Yaw.

$$R = \begin{bmatrix} C_\phi C_\psi - C_\theta S_\phi S_\psi & -C_\psi S_\phi - C_\phi C_\theta S_\psi & S_\theta S_\psi \\ C_\theta S_\psi S_\phi + C_\phi S_\psi & C_\phi C_\theta C_\psi - S_\phi S_\psi & -C_\psi S_\theta \\ S_\phi S_\theta & C_\phi S_\theta & C_\theta \end{bmatrix} \tag{2}$$

For a given vector $\vec{v}$ in the body frame, the identical vector is given by $R\vec{v}$ in the inertial frame.

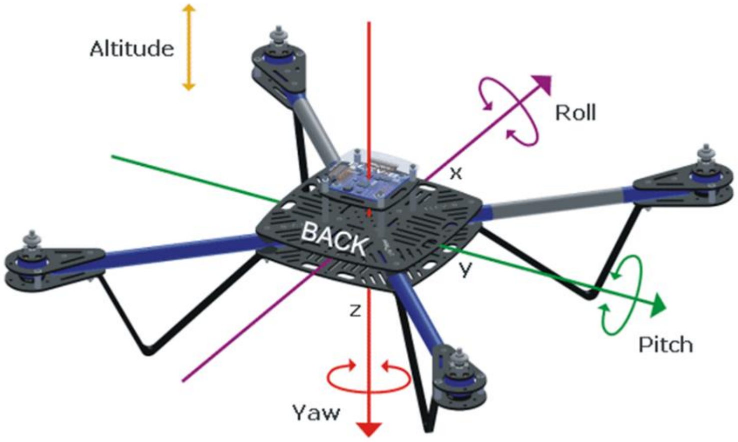

Fig 2. Center of Gravity of UAV.

For a given vector $\vec{v}$ in the body frame, the identical vector is given by $R\vec{v}$ in the inertial frame.

2.2 Motion Controller

The motivation behind deriving a numerical model of a UAV is to help with creating controllers for physical UAVs. The inputs to our framework comprise of the angular velocities of every rotor since everything we can control is the voltages over the motors. Note that we just utilized the square of the angular velocity, ω_i^2 and never the angular velocity itself, ω_i. For notational effortlessness, let us present the inputs $\gamma_i = \omega_i^2$. Since we can set ω_i we can obviously set γ_i also. With this, we can compose our system as a first order differential equation in state space. Suppose x_1 be the position in space of the UAV, x_2 be the UAV linear velocity, x_3 be the Roll, Pitch, and Yaw angles, and x_4 be the angular velocity vector. (al of these are 3-vectors.) With these being our state, we can compose the state space equations for the development of our state.

$$\dot{x}_1 = x_2 \tag{3}$$

$$\dot{x}_2 = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{1}{m} R\mathrm{T}_B + \frac{1}{m} F_D \tag{4}$$

$$\dot{x}_3 = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & C_\theta S_\phi \\ 0 & -S_\phi & C_\theta C_\phi \end{bmatrix}^{-1} x_4 \tag{5}$$

$$\dot{x}_4 = \begin{bmatrix} \tau_{\emptyset} I_{xx}^{-1} \\ \tau_{\theta} I_{yy}^{-1} \\ \tau_{\psi} I_{zz}^{-1} \end{bmatrix} - \begin{bmatrix} \frac{I_{yy}-I_{zz}}{I_{xx}} \omega_y \omega_z \\ \frac{I_{zz}-I_{xx}}{I_{yy}} \omega_x \omega_z \\ \frac{I_{xx}-I_{yy}}{I_{zz}} \omega_x \omega_y \end{bmatrix} \quad (6)$$

2.3 PID Control

A PID control is a PD control with another term included, which is corresponding to the integral of the process variable. Including an integral term makes any remaining steady-state error to develop and enact a change, so a PID controller ought to have the ability to track our path (and stabilize the UAV) with an essentially smaller steady-state error. The equations stay “identical to the ones displayed in the PD case, but with an extra term in the error”:

$$e_{\phi} = k_d \dot{\phi} + k_p \int_0^{\mathrm{T}} \dot{\phi} dt + \ k_i \int_0^{\mathrm{T}} \int_0^{\mathrm{T}} \dot{\phi}\ dt\ dt \quad (7)$$

$$e_{\theta} = k_d \dot{\theta} + k_p \int_0^{\mathrm{T}} \dot{\theta} dt + \ k_i \int_0^{\mathrm{T}} \int_0^{\mathrm{T}} \dot{\theta}\ dt\ dt \quad (8)$$

$$e_{\psi} = k_d \dot{\psi} + k_p \int_0^{\mathrm{T}} \dot{\psi} dt + \ k_i \int_0^{\mathrm{T}} \int_0^{\mathrm{T}} \dot{\psi}\ dt\ dt \quad (8)$$

However, PID controls come with their own inadequacies. One trouble that ordinarily happens with “a PID control is known as integral windup. In some cases, integral wind-up can cause stretched oscillations instead of settling. In other cases, wind-up may indeed cause the system to become unstable, instead of taking longer to reach a steady state”, as shown in **Fig ure** below.

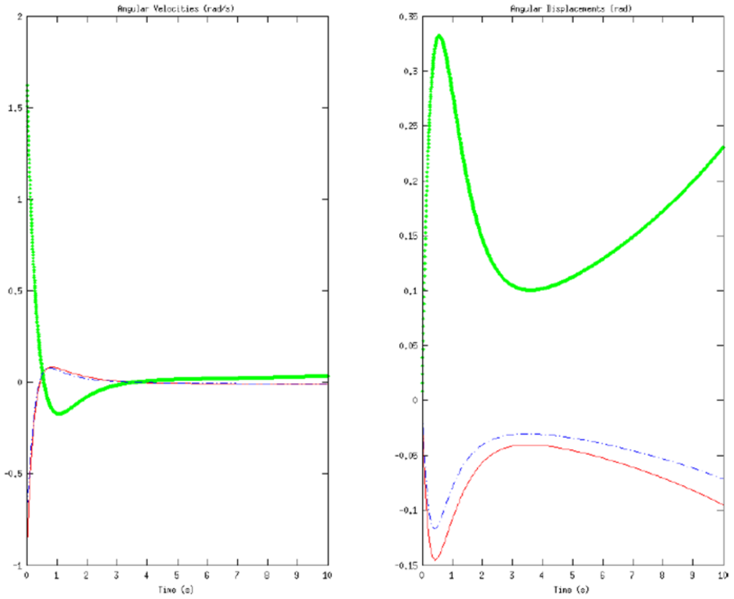

Fig. 3. PID Controller.

If there is a large trouble in the process variable, this large trouble is integrated over time, becoming a still bigger control signal (due to the integral term). However, even once the system stabilizes, the integral is still big, therefore making the controller overshoot its objective. It may then start "a series of dieing down oscillations, become unstable, or basically take an incredibly long time to reach a steady state. In order to avert this, we disable the integral function until we reach something near to the steady state. When we are in a controllable region near the desired steady state, we turn on the integral function, which pushes the system towards a minimal steady-state error", as shown in **Fig ure 4**.

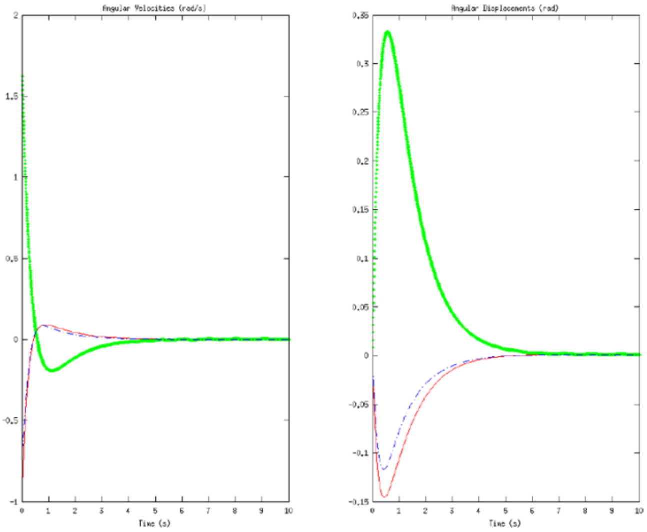

Fig.. 4. With a properly implemented PID, we achieve an error of approximately 0.06 after 10 seconds.

3 Object detection System

In order to establishment a connection between UAV and the earth station (desktop or laptop) we used TeamViewer application. The TeamViewer have many advantages over using real IP. One of them it is free to use while the real IP providers requires fees per period. Also the TeamViewer is more easer in installation and use rather than real IP.

The earth station controlled the Raspberry Pi by running its operating system directly through TeamViewer and run the GUI of our UAV with all its instructions.

For more features, the UAV able to tracking objects by using Raspberry Pi camera and based on the BGR color moment calculations.

3.1 Color detection in OpenCV and Python

Color detection is the process of finding certain color and extracts it from its surroundings in order to detect objects having the same colors. The process in OpenCV is constructed using (cv2.inRange) function which is built in function and this function detects colors in a rage between lower and upper values that are provided for certain color range like blue for example in our work, in the following **Figure 5** our detected object based on color.

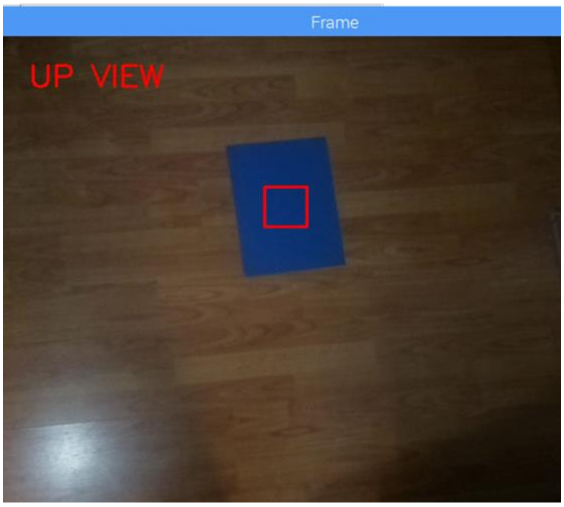

Fig. 5. Detected object based on color.

In color detection it is recommended to convert the captured image from 'BGR' color range to HSV color range because it is more easer to isolate colors from each other's.

In numerical representation in our code:

lower = np.array([76,31,4]) represents the lower value for the color that we intended to detect.

upper = np.array([210,90,70]) represents the upper value for the color that we intended to detect.

thresh = cv2.inRange(blur, lower, upper) this part will detect the color in the image captured by camera that is limited between the lower and upper values of that is provided by us.

Then we will find the contours in the image or frame under test by the following line in our code:

contours,hierarchy=cv2.findContours(thresh,cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)

After that we will find the contour with the maximum area and we will consider it as our goal as shown in the following lines in code; finding contour with maximum area and store it as best_cnt:

```
max_area = 0
best_cnt = 1
for cnt in contours:
    area = cv2.contourArea(cnt)
    if area > max_area:
        max_area = area
        best_cnt = cnt
```

After that we will find the centroid of the best contour and we will draw a box surrounding it as shown in previous **Figure 5**.

```
M = cv2.moments(best_cnt)
cx,cy = int(M['m10']/M['m00']), int(M['m01']/M['m00'])
```

4 System simulation

The simulation process is made using Sims cape multi body, before that CAD model should be imported from one of the mechanical 3d construction design programs, like solid works or others The quad copter consists of many collected items which will represent the full Drone body like shown in previous figure.

The importance of importing the drone as separated parts forming the full body is to control some of these parts and leaving others like rotating the motor shaft and keeping the motor body stable same as in reality.

The process of controlling the Quad copter or making flight controller for it consists of many stages starting with inputs which are coming from the sensors which represents the values of (Roll, Pitch, Yaw, Altitude) as shown in **Figure 6** below:

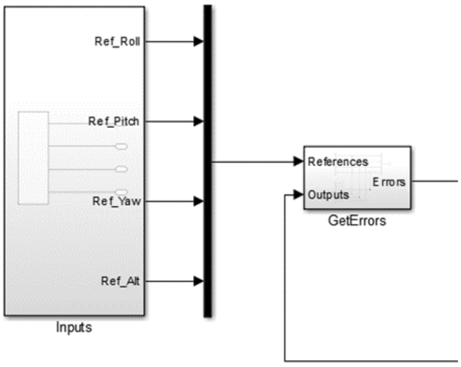

Fig. 6. The sensor outputs

The main benefit of sensor values is to give feed back to the controller to check the level of the drone axis, for example if we have the value of Pitch is (-22) it means that the drone is turning to right side and if we will leave it turning for unwanted period of time that means it will fall down, so the controller is responsible of returning the drone to the horizontal plane to prevent it from falling down.

The same procedure will be applied for (Yaw, Roll and Altitude) to keep the drone stable in its flight.

The most important issue is that the input coming from the IMU which will represent the values of (Yaw, Roll, Altitude), these values should be accurate to provide the flight controller with suitable values to insure the best flight, the inputs are simulated by using the input block shown in figure (4.2) above.

The next step in the flight controller procedure is to calculate the error for (Yaw, Roll, Pitch and Altitude), error means the difference between the desired values and the measured ones, this step is made by the (GetErrors) block drown in **Figure 7** above.

After that the error values will be injected to flight controller in order to be evaluated and to be corrected as shown in figure 6 below.

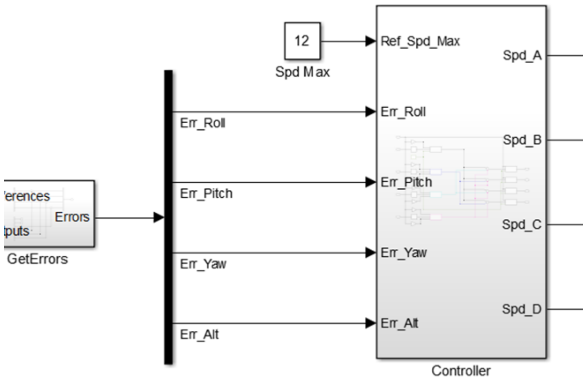

Fig. 7. The injected values from error calculating block to the flight controller.

Inside the flight controller, a correcting process will be made to reduce the error for (Yaw, Roll, Pitch and Altitude) smoothly using PID algorithm for each value separately. The main purpose for applying simulation for drone is to estimate the correct values for (P gain, I gain, D gain) in a way that prevents the drone from doing extreme actions or to go unstable during flight.

To go further in our explanation we will show the inner circuits inside controller to show the hall procedure made by our system, as shown in **Figure 8** below.

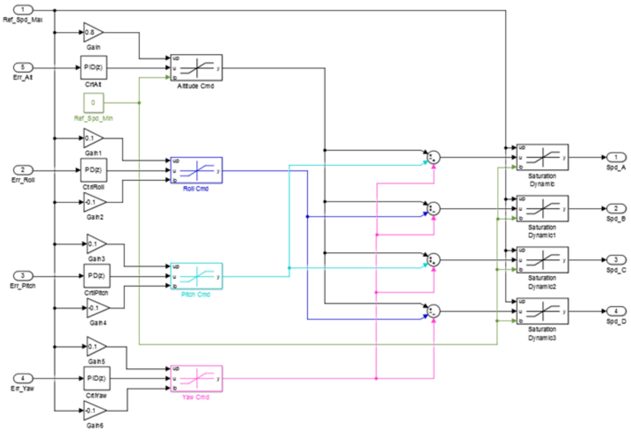

Fig. 8. The PID controller for (Yaw, Roll, Pitch and Altitude).

To explain the operation of the circuit we will start with the first section as shown in **Figure 9** below:

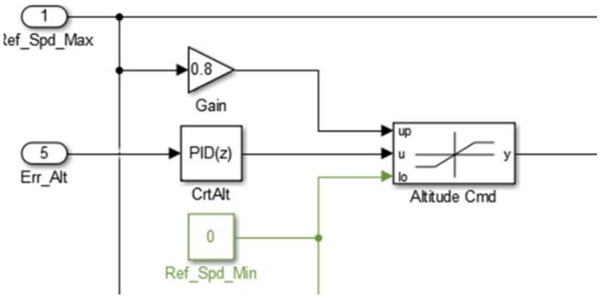

Fig. 9. First section inside controller.

The PID controller for (z) means the PID for altitude; we can see that PID controller takes its input from the error signal (Err_Alt) which represents the difference between the actual altitude and the desired one, inside PID controller a PID control will be applied by selecting the values related to it as follows:

Proportional (P): 4.5
Integral (I):10
Derivative (D): 1

The output of PID controller is shown in **Figure 10** below and as we can see that both of its, stability and transient time is suitable (the most important goals of PID control), we will discuss the tuning effect on behavior later on.

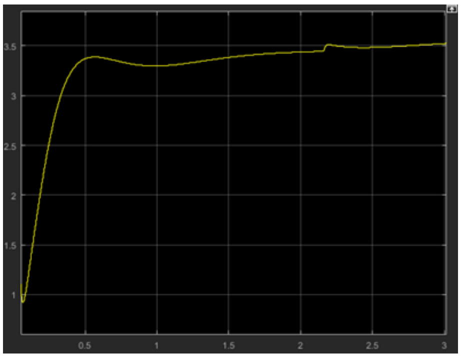

Fig. 10. PID (z) output behavior.

The purpose of adding Altitude Cmd is to limit the boundaries of PID (z) control between the 0.8*(Ref_Spd_Max) as an upper limit and (Ref_Spd_Min) which represent the maximum reference speed and minimum reference speed, which means preventing the drone from reaching undesired speed. After that the (Saturation Dynamic) block also prevents the drone from exceeding the dynamic boundaries.

Finally the corrected values will be send out as a decision to Quad motors through terminals(Spd_A, Spd_B, Spd_C, Spd_D) to (Quadcotor 3D Model) block to reposition the drone to the correct level and frame.

After starting the simulation we can draw the trajectory of drone by drawing the XY graph as shown in **Figure 11** below:

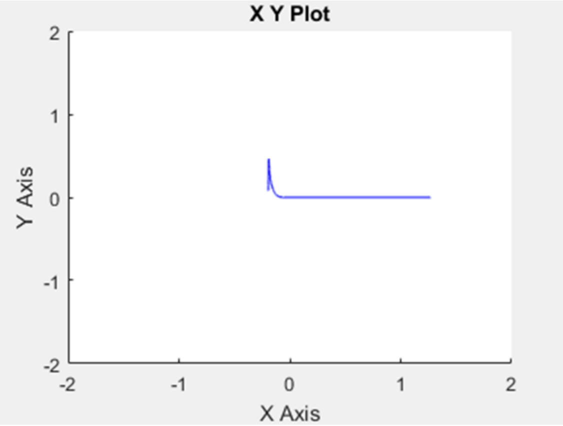

Fig. 11. Drone trajectory.

In the real time operation of quadcopter the commands operating it comes from remote control unit but in simulation in Simulink these commands can be applied from signal builder block in Simulink as shown **Figure 12** below:

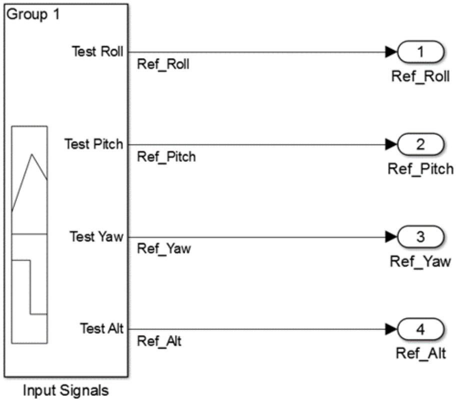

Fig. 12. Signal builder as an input signals.

After starting simulation a 3D drone will be shown on mechanics explorer window as shown in **Figure 13** below:

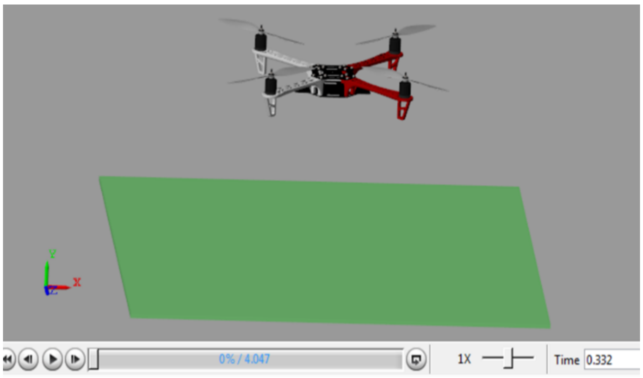

Fig. 13. Mechanics explorer's window during simulation.

5 Conclusion and future work:

The new generation of Wireless Sensor Networs, that is known as the the Internet of Things (IoT) enables the direct connection of physical objects to the Internet using microcontrollers. In this proposal system, an UAV system that is controlled using internet connection which gives capability for infinite distance control and detect certain objects using their colors is devoped based on internet of things Technology and PID controller. As seen from the minssion results that there are infinite probability for changing PID gain values so the tuning of the right values need to be tried to get a small transient period and less oscillation but there will be limitations because the drone will response slowly to the process of control.

In the futer work it is recommended to use Fuzzy Logic in after recognition step in order to control the movment of object toward target.

References

[1] Brien, O., Michael, H., (2016). "The Internet of Things", Journal of Internet Law, 19 (12) : 1-20.

[2] Bughin, J., Chui, M., and Manyika, J., (2015). "An executive's guide to the Internet of Things", Mckinsey Quarterly, 4 : 92-101.

[3] Velez, F., J., Nadziejko, A., Anders, L., Oliveira, S., Rodrigues, T., Costa, V., Duarte, M., Silva, F., and Gomes, J., (2015). "Wireless Sensor and Networking Technologies for Swarms of Aquatic Surface Drones", IEEE 82nd Vehicular Technology Conference. 10(1109):739-793.

[4] Zahugi, E., Mohamed, H., Shanta, Mohamed, M., and Prasad, T.V., (2012). "Design of Multi-Robot System for Cleaning up Marine Oil Spill", International Journal of Advanced Information Technology, 2(4):192-198.

[5] Burange, W., and Misalkar, H., D., (2015). "Review of Internet of Things in development of smart cities with data management &amp; privacy", International Conference on Advances in Computer Engineering and Applications, 10(1109):164-193.

[6] Spalazzi, L., Taccari, G., and Bernardini, A., (2014). "An Internet of Things ontology for earthquake emergency evaluation and response", International Conference on Collaboration Technologies and Systems (CTS), 20(14):686-699.

[7] Jiuyan, C., Bin, L., Q. and Fang, Y., (2013). "A precision agriculture management system based on Internet of Things and WebGIS", (2013). 21st International Conference on Geoinformatics, 66(2):61-73.

[8] D. Nield, “Boeing’s latest patent reveals a drone that can transform into a submarine”, ScienceAlert. http://www.sciencealert.com/boeings-latest-patent-reveals-a-drone-that-can-transform-into-a-submarine , 2016.

[9] Economic Research Office, Information and Communications in Japan, Ministry of Internal Affairs and Communications (MIC), Japan. http://www.soumu.go.jp/johotsusintokei/whitepaper/eng/WP2011/index.html, 2011.

[10] Qiantori, A., Sutiono, A., B., Hariyanto, H., Suwa, H., and Ohta, T., (2010). “An emergency medical communications system by low altitude platform at the early stages of a natural disaster in Indonesia” , J. Med. Syst., 36(1):41–52.

[11] Rogers, J., (2015). “How drones are helping the Nepal earthquake relief effort,” FoxNews.Com. http://www.foxnews.com/tech/2015/04/30/how-drones-are-helpingnepal-earthquake-relief-effort.html , April 2015.

[12] R. Debra and C. McCullough, Unmanned Aircraft Systems (UAS) Guidebook in Development. http://cops.usdoj.gov/html/dispatch/UAS_Guidebook_in _Development.asp , 7 Mar 2016

[13] S. W. Loke, “The Internet of Flying-Things: Opportunities and challenges with airborne fog computing and mobile cloud in the clouds,” Dept. Comput. Sci. Inf. Technol., La Trobe Univ., Melbourne VIC, Australia, Jul. 2015, https://arxiv.org/abs/1507.04492

[14] Brien, J., O., Baran, D., G., and Luu, B., B., (2006). “Ad hoc networking for unmanned ground vehicles: Design and evaluation at command, control, communications, intelligence, surveillance and reconnaissance on-the-move,” ARL, Adelphi, MD, USA, Army Res, 39:91-101.

[15] IAFC (International Association of Fire Chiefs). (Jan. 2014). Use of Unmanned Aerial Vehicles in Public Safety Emergency Response, Adopted by IAFC Board of Directories. http://www.iafc.org/IAFC-position-Use-of-Unmanned-Aerial-Vehicles-In-Emergency-Response

[16] Bolkcom, C. (2004, June). Homeland security: Unmanned aerial vehicles and border surveillance. Library of Congress Washington DC Congressional Research Service.

[17] M. T. McCaul, (2012). “Using unmanned aerial systems within the homeland: Security game changer?” House Committee Homeland Security, Washington, DC, USA, Tech. Rep, 11(2):10-17.

[18] W. Staff, “Fighting forest fires before they get big—With drones,” WIRED, Jun. 2015. http://www.wired.com/2015/06/fighting-forest-fires-get-big-drones

[19] J. K. Hart and K. Martinez, “Environmental sensor networks: A revolution in the earth system science?” Earth Sci. Rev., vol. 78, nos. 3–4, pp. 177–191, Oct. 2006.

[20] P. Dockrill, “Facebook is preparing its Internet-beaming drone for maiden launch,” ScienceAlert. Accessed on Mar. 7, 2016. http://www.sciencealert.com/facebook-is-preparing-its-internet-beaming-drone-for-maiden-launch.

[21] D. Sahota, “Internet.org building drones to connect remote communities,” Telecoms.com, Mar. 2014. [22] T. Skinner, “Nokia and Du launch network testing drones in Dubai,” Telecoms.com, Jul. 2015. http://telecoms.com/430141/du-and-nokia-launch-network-testing-drones-in-dubai/ .

[23] G. Kimchi et al., “Unmanned aerial vehicle delivery system,” U.S. Patent 20 150 120 094, Apr. 2015.

[24] D. Lee, “Google plans drone delivery service for 2017,” BBC News. http://www.bbc.com/news/technology-34704868 .

[25] Amazon, “Amazon prime air” , http://www.amazon.com/b?node=8037720011