Macrothink Institute™

Network Protocols and Algorithms
ISSN 1943-3581
2017, Vol. 9, No. 3-4

Unmanned Aerial Vehicular System for Greenhouse Gas Measurement and Automatic Landing

Henok Ali, Momen Odeh. Ahmed Odeh, Ali Abou El-Nour, and Mohammed Tarique
Department of Electrical Engineering
P.O. Box 2202, Fujairah,
United Arab Emirates (UAE)
E-mail: m.tarique@ajman.ac.ae

Received: October 10, 2017 Accepted: December 22, 2017 Published: December 31, 2017

DOI: 10.5296/npa.v9i3-4.12319 URL: http://dx.doi.org/10.5296/npa.v9i3-4.12319

Abstract

This paper presents a reliable and low cost greenhouse gas measurement system. The system mainly consists of an unmanned aerial vehicle (UAV), a set of calibrated sensors, a wireless system, and a microcontroller. The system can measure the concentration of greenhouse gases namely carbon dioxide (CO_2), methane (CH_4), and ozone (O_3) at different altitudes. It can also measure temperature, humidity, and atmospheric pressure. The system is able to send data to a remote monitoring station. The UAV is equipped with image processing based navigation and landing system so that it can land autonomously on a designated place. To ensure safe landing the system uses a specially designed parachute. This paper also presents some data generated by the system.

Keywords: Arduino, global warming, greenhouse gases, measurement, navigation, landing, sensors, UAV, wireless, sensors.

1. Introduction

Environmental pollution has existed since industrial revolution and has become an issue of serious international concern in the recent years. Researchers, from academia and industry, are investigating the causes and effects of environmental pollution. The toxic substances (i.e., pollutants), released into the ecosystem, are polluting the environment. As the world's population continues to grow, so does these toxic substances. Researcher and scientists have categorized these toxic substances as soil, water, and air pollutants.

Soil pollutants affect the Earth's surface. Heavy metals including cadmium, chromium, pesticides, organic chemicals, and radioactive materials cause soil pollution. Water pollutants affect the Earth's water sources. Mercury, nitrates, phosphorous, fecal coliform and bacteria cause water pollution. Air pollutants affect the Earth's atmosphere. Ozone, particulate matter, carbon monoxide, nitrogen oxides, sulfur dioxide and lead mainly cause air pollution. Among these pollutions, mentioned above, air pollution has become a serious concern now.

Until now, air pollutants are causing the most harmful effects on our environment. Air pollutants enter into the atmosphere through industrial processes including generation of heat and power, treatment and disposal of solid wastes, and emission from vehicles. Among these, emission from vehicles alone generates approximately 60% of all air pollutants. Researchers and scientists have named carbon dioxide (CO_2), methane (NH_4), nitrous oxide (N_2O), water vapor (H_2O), and Ozone (O_3) as major air pollutants. These pollutants are also called greenhouse gases (GHGs). The GHGs trap long wave radiation emitted by the Earth's surface and make it habitable [1]. Among these GHGs, the CO_2 worries us the most. The unprecedented consumption of fossil fuels is generating a huge amount of CO_2 and is polluting the atmosphere at a very alarming rate. In the recent years, the GHGs are trapping more heat on the Earth's atmosphere [2] and are causing global warming. The global warming is affecting our existing ecosystem and biodiversity [3,4,5]. According to the scientists, the Earth's surface temperature is rising and it could exceed all historical values by 2047 [6, 7, 8]. Hence, it is very important to monitor the temporal and spatial variation of the concentration of these GHGs in the atmosphere. In this work, we present an Unmanned Aerial Vehicle (UAV) based greenhouse gas measurement system to monitor the levels of GHGs in the environment.

Numerous greenhouse measurement systems have been reported in the literatures. These systems use different methodologies. Measuring greenhouse gases using UAV is one of them. The major limitation of these existing systems is that they use high tech UAVs equipped with sophisticated sensors and instrument. Hence, these systems are quite expensive. To overcome this limitation we use a cheap Quadcopter (Phantom 3 Advanced) in this work. The Phantom 3 Advanced is a small aerial vehicle with four propellers attached to four motors located at the cross frame as shown in Fig.1. These four motors control the UAV's motion. In this work, we equip the UAV with a set of gas sensors and microcontroller. We also develop algorithms for controlling the flight. The present system is able to measure the concentration of the GHGs namely CO_2, CH_4, and O_3 at different altitudes (allowed by the local aviation authority). The system is also able to send the data to a remote monitoring station. The system is able to store the data in an on-board SD memory card. We have developed a landing and navigation system

for this system so that the UAV can land in a specific landing place after completing the mission. To ensure safe landing we have designed a parachute for the UAV.

Figure 1. The Quadcopter system.

The rest of the paper is organized as follows. Section 2 presents some related works. Section 3 presents the design of flight controller. Section 4 presents the greenhouse gas measurement system. Section 5 presents experimental results. Section 6 and section 7 present the landing system and parachute design respectively. Section 8 concludes the paper.

2. Related Works

Now-a-days, the UAVs are being used in many applications including public safety, disaster management, scientific research, agriculture, and environment [9,10]. In this section, we present some recent UAV based greenhouse gas measurement systems that are available in the literatures. This section also presents the limitations of the existing related works and discusses the advantages of our present system over other existing systems.

Raymond Hunt et. al. present a radio-controlled aircraft based plantation monitoring system in [11]. The main target of this system is to monitor the growth of plantation in a cornfield. The aircraft, equipped with an infrared camera, flies over the cornfield and captures images of the same. Based on these images, the system calculates NDVI (Normalized Difference Vegetation Index). The NDVI measures the growth rate of the plantation.

András Molnár1 presents a comprehensive system in [12]. It consists of a carrier device, measurement system, and software program. The system monitors a small area at low altitude. The system is able to detect pollution level over the time. It is also capable of measuring and recording air quality parameters including humidity, temperature, dust, radiation, and chemical pollution with a high geographical precision. The system is also able to send data to a ground monitoring system. Based on the data the system generates a 3D map illustrating the distribution of air pollution level over the observation area.

Acevedo et.al. use a fleet of heterogeneous UAVs, with limited communication range, in [13]. The main goal is to detect a pollution source. To fulfill this goal the system divides the area into small areas and the UAVs fly over the areas. Once a source of pollution is detected, the whole fleet informs each other about it. A sufficient condition has been derived for this purpose so that the whole fleet can share information regardless of propagation constraints.

Kingston D. et al. present another similar work in [14]. In this work, a decentralized multiple-UAV approach has been used for perimeter surveillance. This perimeter surveillance monitors forest fire. Two basic perimeter topologies, circular and linear, have been investigated in this work. The proposed system is able to gather data at all points on the perimeter and transmit that data to a base station for further analysis. To ensure accuracy of the surveillance the UAVs are spread uniformly along the perimeter.

Maza I. et. al. present a similar surveillance system in [15]. The system is consisting of autonomously distributed UAVs, wireless sensor/actuator networks, and ground camera networks. The system architecture is designed based on task allocation, conflict resolution, task decomposition, and sensor data fusion. The system operates in disaster management scenario. The system is made suitable for four different missions namely fire confirmation, surveillance, fireman tracking, and load transportation.

Viguria A. et. al. present another similar work in [16] for emergency management application. The system coordinates aerial and ground robots for surveillance. The authors have presented a distributed market-based algorithm, called S + T. By using this algorithm, the robots can provide transportation and communication relay services dynamically to other robots during an emergency mission. The test results show that the heterogeneous robots (aerial and ground) are able to detect fire by cooperating with each other. The system is also able to extinguish the fire.

Laliberte A.S. et. al. use two UAVs in [17] for rangeland mapping, assessment, and monitoring. These UAVs, equipped with GPS guidance capability, can detect a pollution source over a large area. One of these UAV is a modified model airplane. It is able to fly along preloaded waypoints and acquire images with a digital camera. The other UAV has full autonomous flight capability and is equipped with color video and digital cameras. These two UAVs provide a data file containing GPS and elevation for each image. Both systems acquire high quality and high-resolution images at 150m flying height.

Sidek O. et. al. present a work in [18] that is similar to our present work. They present a prototype system for monitoring and computing the GHGs with an UAV. The system uses the ZigBee technology to transfer the data. The UAV also acts as a router and it sends data again to a data logger. In this system, an ATMEGA328P microcontroller is used. This microcontroller is programmed to measure the concentrations of CO_2, O_2, temperature, and humidity at different altitudes. The system measures all these environmental parameters on real time basis and stores data in SD card for every 30 seconds interval.

Malaver Rojas et. al. propose a solar powered UAV system in [19]. The system uses a Wireless Sensor Network (WSN) and it can measure greenhouse gases in agricultural lands. The system uses a generic gas sensing system using infrared sensors. The major advantage of

this system is that it is eco-friendly because of its solar power source. The system is also associated with a data management system, which stores, analyzes, and shares the information with operators and external users. The test results show that the system is able to collect, store, and transmit data in real time.

Chris Hugenholtz, and Thomas Barchyn present a small drone in [20] to measure atmospheric concentration of CH_4. This small drone is equipped with a gas detector to measure CH_4 plume from a controlled source. The field-test results indicate the system is able to locate accurately the source of CH_4 emission. The major limitation of this system is its high cost.

Juan Jesús Roldán et. al. present another similar work in [21]. They present the design, construction, and validation of greenhouse gas monitoring system. The system consists of a sensory system placed on a four-rotor mini-UAV. The system measures temperature, humidity, luminosity, and CO_2 concentration. It also produces a map based on these parameters.

One of the common limitation of the above mentioned systems is that they all use high tech UAVs equipped with sophisticated sensors and instruments. Hence, they are very expensive. In addition, these systems are difficult to install and configure. To overcome these limitations we present a novel greenhouse gas monitoring system in this work. We use programmable UAV in this work. The UAV is able to navigate autonomously. After completing the mission, the UAV is also able to land on a designated landing place. To prevent damages, while landing, we have designed a special parachute for the UAV.

3. Flight Controller Design

The main component of the present system is a Quadcopter as mentioned above. It has four propellers attached to four motors located at the cross frame. The speeds of these motors are varied to control the direction of flight. In this work, we develop a program for an Arduino microcontroller to control the flight. The flowchart of this program is shown in Fig.2.

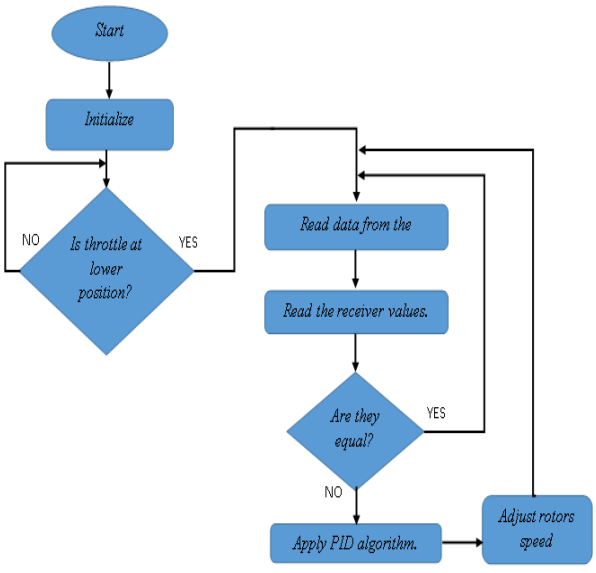

Figure 2. The Quadcopter Arduino-based flight controller.

We adopted two options to control the flight. In the first option, we design the flight controller by using a PID controller as shown in Fig.2. We tune the PID controller (see Fig. 3) by the following algorithm.

1. Set K_i and K_d to zero, and increase K_P.
2. Decrease the value of K_P by 50%.
3. Increase the K_i until the Quadcopter starts vibrating.
4. Decrease the value of K_i by 50%.
5. Increase the value of K_d until the Quadcopter becomes stable.
6. Decrease the value of K_d value by 25%.

However, the tuning procedure mentioned above makes the Quadcopter to oscillate. To improve the flight performance we convert the PID controller into a PD controller by setting the K_i gain to zero. The PD controller takes derivative from the system's output instead of the error signal and hence it can cope with sudden change in the error signal. The PD results in reducing the oscillation of the UAV during takeoff.

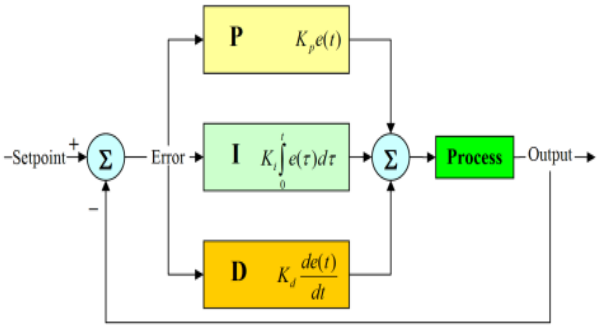

Figure 3. The PID Controller [22].

In the second option, we use the MultiWii [23] software to control the flight. The MultiWii is a general purpose, free, and open source program to control flying RC models. Initially, the MultiWii is introduced on a Wii Motion Plus extension and an Arduino Pro mini board. From a very simple, cheap, and minimalist flight controller the MultiWii has now matured and it supports all expected features including GPS navigation. The programming environment of the MultiWii is shown in Fig.4. In order to build the MutiWii flight controller we follow the steps mentioned below.

1. Start the MultiWii program.
2. Select the port (to connect with Arduino).
3. Press the start button to start reading the values for roll, pitch, and yaw.
4. Calibrate the gyroscope by pressing the CALB_ACC button.
5. Start calibrating the PID gain by pressing the READ button.
6. Change the PID gains and fly the Quadcopter.
7. Keep changing the PID gains until the Quadcopter is stabilized.
8. Open the MultiWii source code with the Arduino platform.
9. Select the com port used by the MultiWii configuration program.
10. Uncomment the multi-rotor module and the gyroscope module.

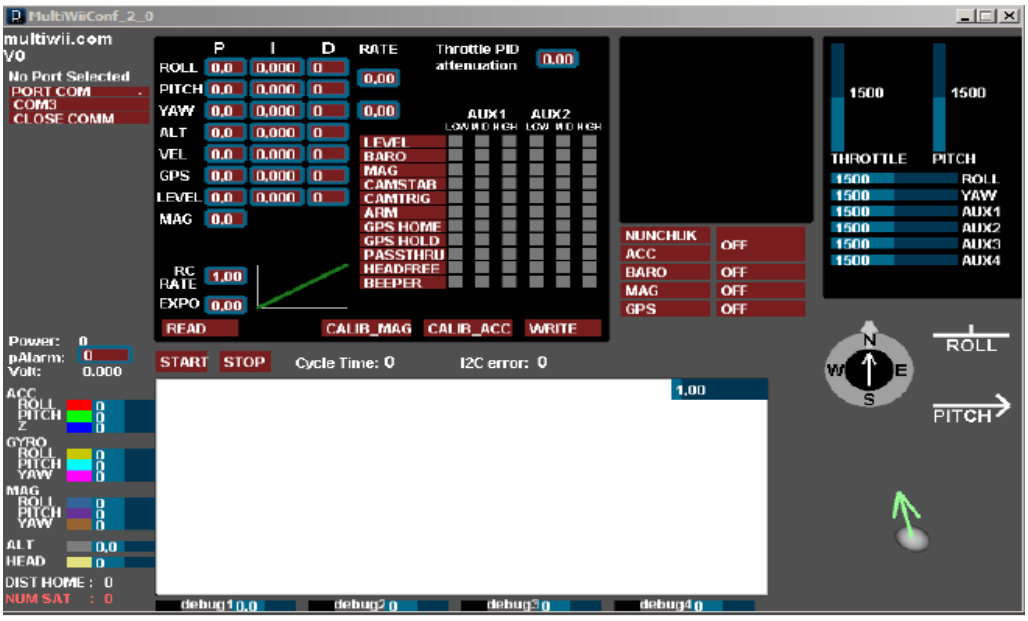

Figure 4. The MultiWii Programming menu.

We carefully select two important parameters namely RC rate and RC expo for MultiWii. The RC rate defines the sensitivity of the Quadcopter's pitch. One needs to adjust this parameter to make the Quadcopter less reactive. The RC expo defines a smoother range at the center of pitch. Compared to the PD controller we achieve a better control of the Quadcopter by using the MultiWii program.

4. Greenhouse Gas Measurement System

In this section, we explain the greenhouse measurement system. The present system uses three gas sensors namely the MQ 135, the M131, and the MQ 4 to measure the concentration of CO_2, O_3, and CH_4 respectively. In addition, we use two more sensors namely the DHT 11 and the BMP 180. The DHT 11 sensor measures temperature and humidity. On the hand, the BMP 11 sensor measures air pressure. To reduce the payload we use compact, low power, and lightweight laser-based sensors. The overall mass of the sensors is around 200 gms including batteries. Each sensor consumes less than 2W of power. The sensors are battery operated and are capable of fully operating for a long time in diverse sensing environments.

To produce accurate data we calibrate the sensors before deployment. We use a power regression analysis for this purpose. We extract points from the sensitivity curves supplied by the sensor manufacturer and then we derive an equation from these points. The system uses this equation to measure the gas concentration. We use a general equation expressed by (1) for regression analysis.

$$\rho = A \times x^{B} \tag{1}$$

,where ρ is the gas concentration in (ppm, or ppb), *A* is a scaling factor, *B* is an exponent, and $x=R_s/R_0$. The R_0 is the sensor resistance in clean room condition. Equation (2) determines the sensor resistance R_s.

$$R_s = \left(\frac{V_{cc}}{V_0} - 1\right) R_L \tag{2}$$

, where V_{cc} is the input voltage, V_o is the output voltage, and R_L is the load resistance (10 KΩ - 47 KΩ). The sensor measures the gas concentration and generates an analog signal. An Analog-to-Digital (ADC) converter then converts the analog signal into a digital signal.

To calibrate all the sensors we follow the same procedure. We will mention the calibration of the MQ 131 sensor here only. To calibrate the MQ 131 sensor, we use the sensitivity curve provided by the manufacturer. The sensitivity curve is shown in Fig.5. By extracting the points from this curve we plot a similar curve by using Excel (see Fig.6). Based on this plot we derive (3).

$$\rho_{ppb} = 24.986 \times \left(\frac{R_s}{R_0}\right)^{-1.289} \tag{3}$$

Similarly, based on the data sheets we derive the expressions, (4) and (5) for the MQ 131 and MQ 4 gas sensors respectively. After measuring all these parameters (i.e., CO_2, CH_4, O_3, temperature, humidity, pressure, and altitude), the system saves data in an onboard SD card module.

$$\rho_{ppm} = 108.6 \times \left(\frac{R_s}{R_0}\right)^{-2.696} \tag{4}$$

$$\rho_{ppb} = 1079.3 \times \left(\frac{R_s}{R_o}\right)^{-2.736} \tag{5}$$

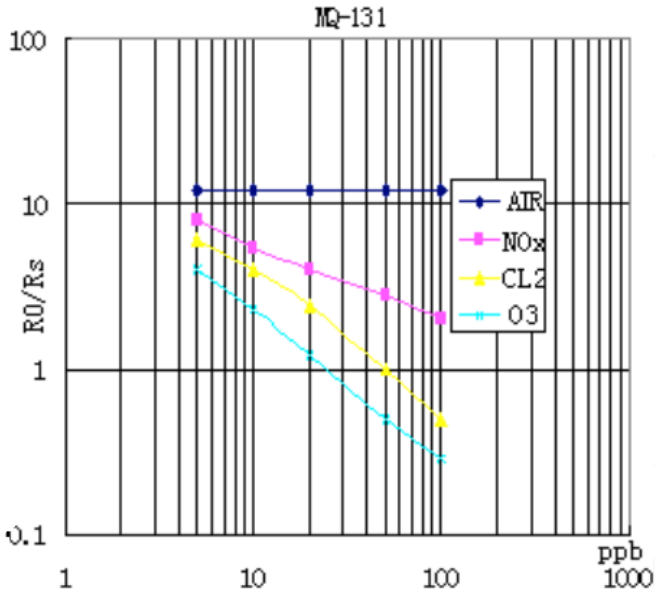

Figure 5. The MQ 131 Sensitivity Curve.

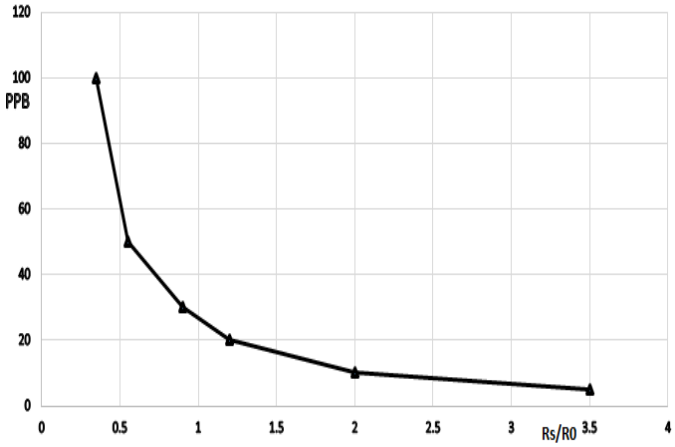

Figure 6. The plot of gas concentration for MQ 131 sensor.

5. The Results

By using the system, we measure the air quality parameters of a local city (i.e., Fujairah) on March 30, 2016. The variations of temperature and pressure over a span of 24 hours are shown in Fig. 7. It is depicted in Fig. 7 that that the temperature varies between 19^0C to 32^0C over the observation period. It attains highest temperature around 2 p.m. and it shows lowest at 4 a.m. However, humidity varies widely over the day. It becomes minimum around 11 a.m. The maximum humidity occurs around mid-night. The air pressure shows minimum (i.e., 1012.5 mb) around 10 a.m. It becomes maximum around 4 p.m.

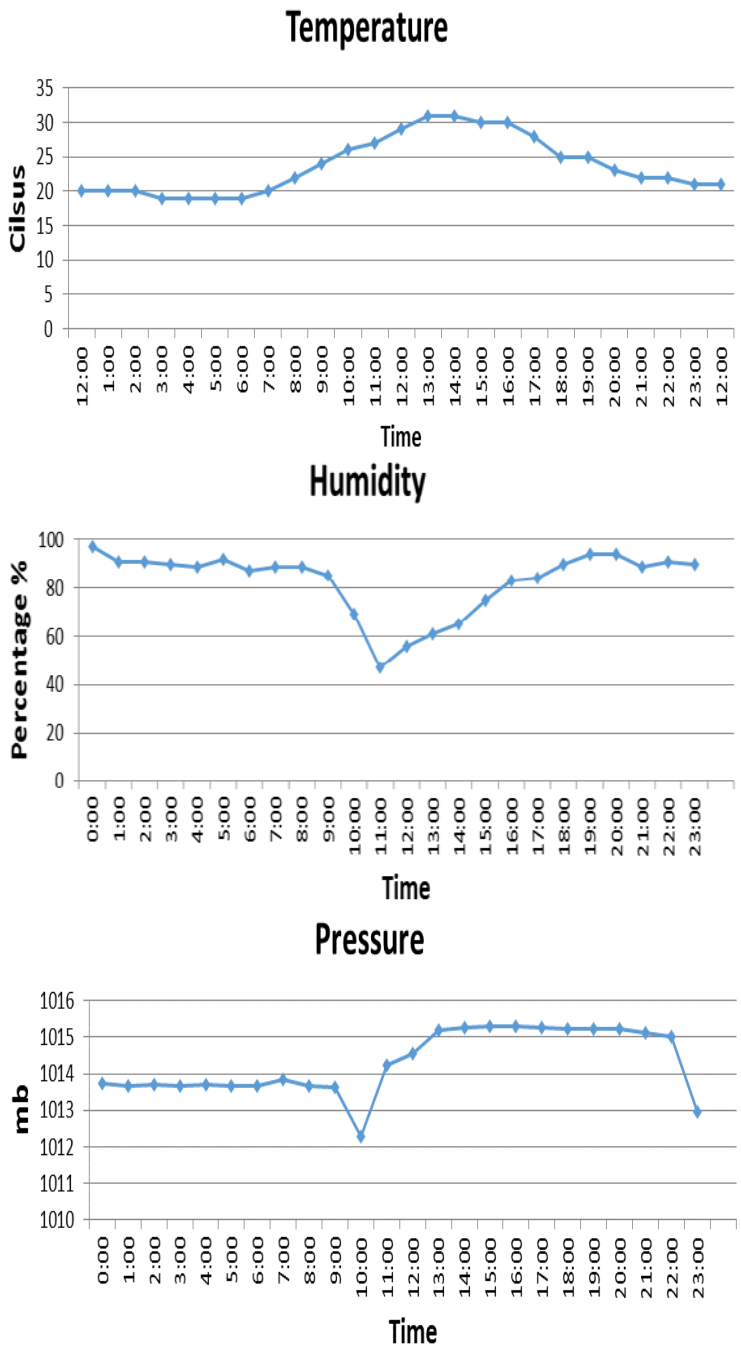

Figure 7. The variation of temperature, humidity, and pressure.

The variation of the greenhouse gases namely CO_2, O_3, and CH_4 are shown in Fig. 8. Among the greenhouse gases, the concentration of CO_2 is the highest during the rush hours (i.e., 4 p.m. to 6 p.m.). It achieves 1350 ppm around 5 p.m. The amount of CO_2 also varies with time and it attains maximum value around 2 p.m. This figure also shows that the concentration of O_3 varies widely with respect to time. It attains a maximum value of 23 ppb during the morning rush hour.

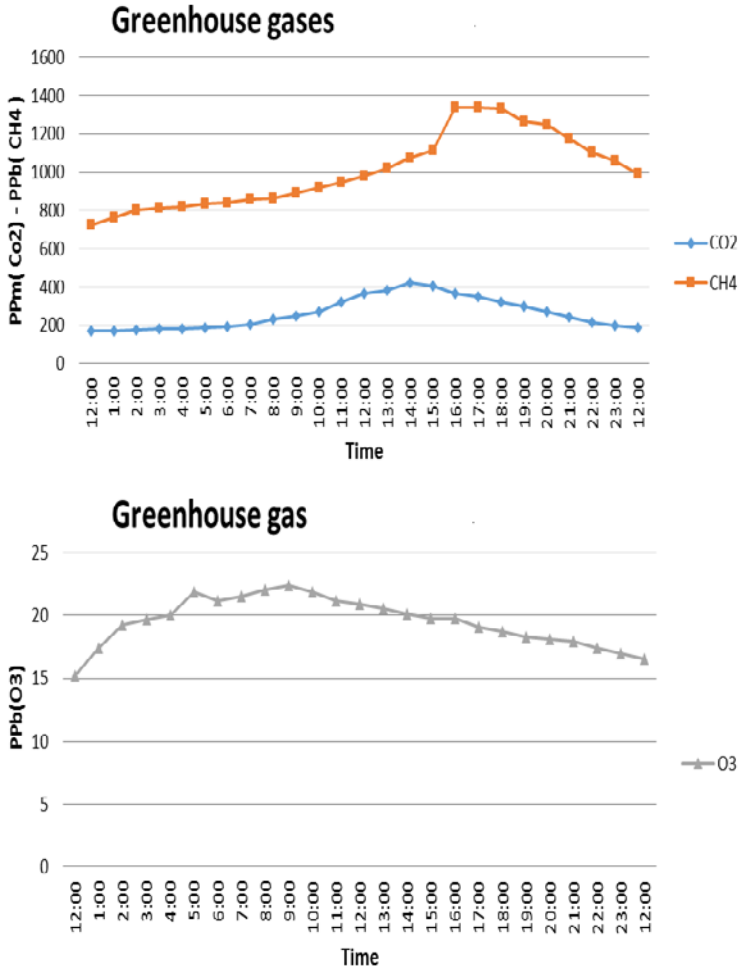

Figure 8. The air quality parameters variation.

We also measure air quality parameters at different altitudes. Table 1 shows a sample data. The data show that the greenhouse gas concentration varies with respect to altitude. It also shows that the concentration of CO_2, O_3, and CH_4 are the highest at the ground level due to heavy traffic. However, the concentration of these gases decrease with the altitude. We limit our effort to measure greenhouse gases up to 200m because of the restriction imposed by local aviation authority.

Table 1. Measurements in University area on Tuesday 4/4/2017 at 10:00 am (latitude: 25.133027, longitude: 56.293559)

Height / Meas	Temp ©	Humidity (%)	Pressure (mb)	CO_2 (ppm)	CH_4 (ppb)	O_3 (ppb)
0 m (Ground)	34.00	15.00	1003.22	349.22	1122.38	146.08
100 m	28.00	18.00	991.21	251.89	944.49	126.42
200 m	27.00	19.00	980.14	173.80	917.32	116.00

6. The Landing System

After measuring the greenhouse gas concentration, the system can land in a designated landing place. We have designed a CanSat for this purpose. By using the CanSat, the system is able to do autonomous landing on a predetermined landing location. The CanSat uses a parachute (which is described in the next section) to control the position of the UAV as well to ensure a safe landing. The landing location has special markers on it. It consists of circular marks as shown in Fig.9.

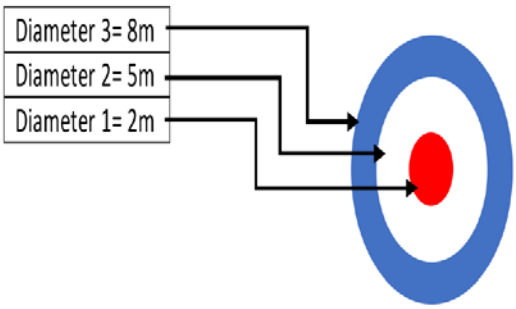

Figure 9. The Color based landing Pad.

In this work, we use a Raspberry PI 2 for designing the landing system. The Raspberry PI is a low cost computer that runs at 900 MHz with an ARM Cortex-A7 CPU and quad-core processor. It has a CSI camera interface that captures images with a resolution of 5.0 MP. In this work, we use this Raspberry PI computer to perform all the tasks related to image processing algorithm and to send control (navigating) commands. We use an open source image processing library called OpenCV [24] for this purpose. We mount a camera at the bottom of the CanSat to capture a top view of the landing location. An Arduino microcontroller navigates the CanSat. The Arduino microcontroller also sends data regarding date, time, altitude, and temperature to a ground station.

For the target tracking, the system uses an image processing algorithm. There are a number of image recognition and identification algorithms available in the literatures. For the sake of simplicity and fast processing we use Kernel based tracker in this project. The Kernel-based trackers are classified into three main classes namely template trackers, multi-view appearance model tracker, and density-based tracker. In this work, we use the template based tracker [25]. This is a color based tracking method that uses range thresholding and contour detection. We use two micro servomotors to steer the CanSat. These servomotors are able to change their shaft from 0-180 degrees by using pulse width modulated signal generated by the Arduino. The shaft extends and shortens the length of a robe connected to the parachute. This change in the length of the robe causes the CanSat to move in a given direction.

An Arduino microcontroller controls these servomotors. The microcontroller also measures the internal temperature of the CanSat using an LM35 temperature sensor. The other functions of the microcontroller include deploy the parachute, parse data from a GPS module, and provide data to the XBee radio module. The GPS module receives data about time, date, and altitude from the satellites. Finally, the microcontroller parses all these data and feeds to an XBee radio module. The XBee radio module is a communication device based on the

IEEE 802.15.14 protocol. The main purpose of this radio module is to transmit all the data to a ground station. Two 3.7V Lithium Ion batteries supply the power for the whole system. Each of these batteries has current rating of 2700 mAh.

We develop a program for finding the target, identifying the target, and sending control commands to the Arduino microcontroller using the GPIO pins of the Raspberry PI. During initialization steps, the library initializes the python's OpenCV library and establishes communication to the Raspberry PI camera. Once the communication is established, the processing sequence starts its recognition and localization. All the processing sequences are summarized in the following steps.

1. Grab RGB frame from the camera.
2. Convert the RGB image into HSV.
3. Create a filtered binary image.
4. Get the contours.
5. Calculate the maximum area.
6. Calculate the position.
7. Send control commands.

Once the frames are captured, the program converts the image from RGB to HSV and the landing pad looks similar to the one shown Fig.10. Then the program performs color filtering and generates a binary image (a black and white image). Fig.11 shows the resultant image after performing color filtration. The next step is to find the contours. The OpenCV provides methods to find the contours in an image. In order to find contours, the program uses the binarized image to obtain the edges.

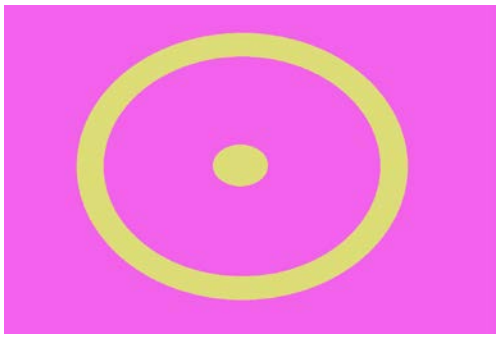

Figure 10. The target after HSV color space conversion.

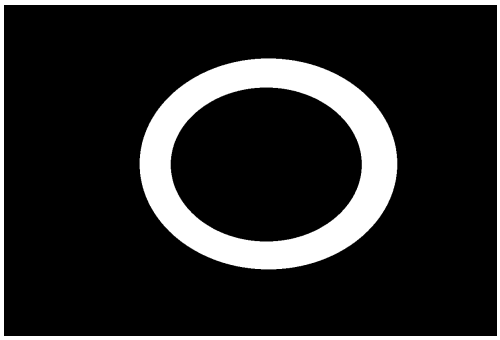

Figure 11. The masked image (for blue color).

There is always a chance for extracting more than one contour while identifying the target landing location. We solve this problem by calculating and choosing the maximum area of the contour. The position of the target is calculated by first drawing a bounding box on the region of interest and then calculating the center of a bounding box by taking the size of the window as a reference (for maximum and minimum values of the position). We use the moments of contour by using Green's formula. We use (6) to calculate the moments.

$$m_{ji} = \sum A(x, y).x^{j}.y^{i} \tag{6}$$

The central moments are calculated by

$$mu_{ji} = \sum_{x,y} A(x, y).(x - \bar{x})^{j}.(y - \bar{y})^{i} \tag{7}$$

, where $(\bar{x}, \bar{y})$ is the mass center and defined by $\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}$

Then we calculate the normalized central moments by

$$nu_{ji} = \frac{mu_{ji}}{m_{00}^{(i+j)/2+1}} \tag{8}$$

We use a set of control commands to locate the target position. This decision is made based on a simple calculation from the vertical and horizontal values of the center of the target. We use four commands for this purpose. A 'left' command is sent when the position on the x-axis is less than 298. A 'right' command is sent when the position on the x-axis is greater than 388 otherwise the target is around the center on the horizontal axis. The other commands are 'front' and 'back'. The 'front' command is sent when the position on the y-axis is less than 258 and the 'backward' command is sent when the position on the y-axis is greater than 218. If all these conditions are fulfilled the system determines the center of the landing place.

We conduct a number of experiments to make sure that the CanSat can land on the predefined target autonomously. Fig.12 shows one of the conducted experiments. The red rectangles in Fig.12 show the parts of a frame that needs to be tracked and landed on. The landing system has been experimented in a real life scenario. The experimental results show that the CanSat can track and land on a predefined colored landing pad. During the landing, the system is able to process up to 25 frames per second. This frame rate is sufficient for making a decent decision for the navigating command.

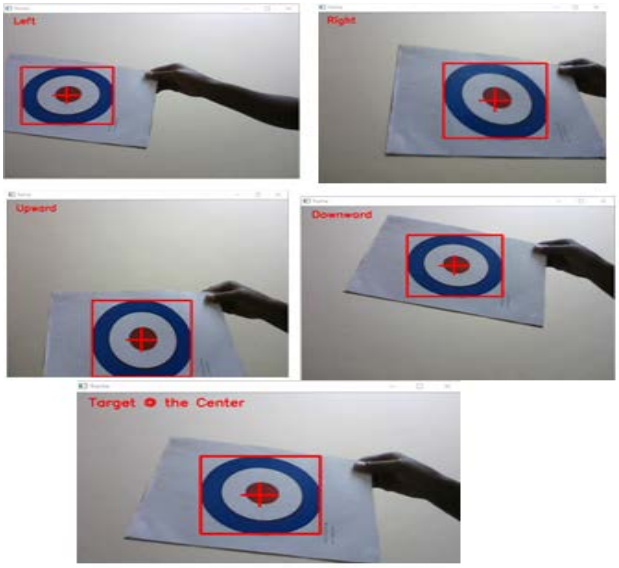

Figure 12. The sequence of frames during the tracking process.

7. The Parachute Design

The system uses a parachute to control the descent speed and to control the path followed by the UAV. We use a strong, thin, and flexible nylon material to make this parachute. To calculate the size of this parachute, we use the specifications mentioned by a local IEEE Industrial design project (IDP) competition. According to the specifications, the CanSat will be at 90m height and it should reach the ground in maximum 60s. For this purpose, we use (9) to determine the descent velocity.

$$v = \frac{90}{60} \quad \text{m/sec} \tag{9}$$

According to the law of gravitational force, we calculate the dragging force for the CanSat by

$$F = mg \tag{10}$$

To descent at constant speed (which equals the terminal velocity), the parachute applies a force called the drag force, F_d opposite to the gravity force, F_g. Hence, we have to satisfy the condition express in (11).

$$F_g = F_d \tag{11}$$

$$mg = 0.5\rho C_w v^2 A \tag{12}$$

,where ρ is local air density (1.255 Kg/m^3), C_w is the drag coefficient of the parachute , *v* is the terminal velocity and A is parachute area. Therefore, we calculate the parachute area by

$$A = \frac{mg}{0.5\rho C_w v_d} \tag{13}$$

Thus, we determine the minimum area = 0.276 m^2, and maximum area = 2.4 m^2. We designed the CanSat according to the competition dimensions (i.e., width and depth=85mm±1mm and height=160mm±1mm). We use hard and strong cardboard as shown in Fig.13 to sustain any kind of crashes or blows that may occur due to landing.

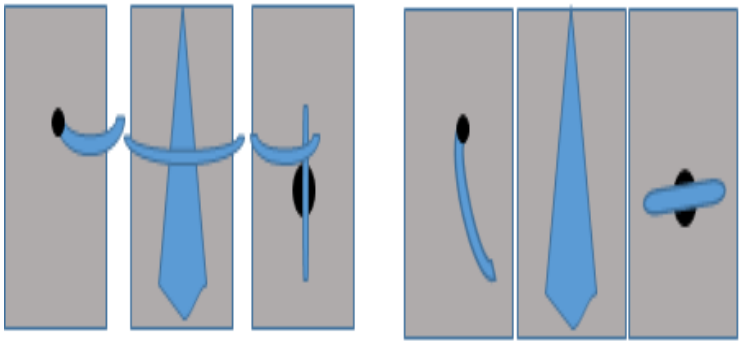

Figure 13. The mechanism of the parachute deployment system.

For controlling the direction or the path of descent, we use two controllers. One is the Raspberry PI for the image processing to locate a target on the ground. The other one is an Arduino Uno to control the servomotors and transmit data into a ground station. We developed two programs for both of these controllers. The flowcharts of these two programs are shown in Fig.14 and Fig. 15 respectively.

Based on the control commands received from the Raspberry PI the CanSat navigates as follows. The system receives the commands as a serial binary data using three pins from both the Raspberry PI and Arduino. The servomotors are controlled by using the voltage levels at the pins. Initially, both the servomotors will be at 90°. When any of the navigating commands is received, one of the servomotors will change its position to 0° in order to shorten the length of the robe attached to a parachute and these results in moving the CanSat into the desired direction. The servomotors need a PMW signal to vary their position. This signal is generated by using the servo library of Arduino. The flowcharts show all other tasks performed by the Arduino microcontroller.

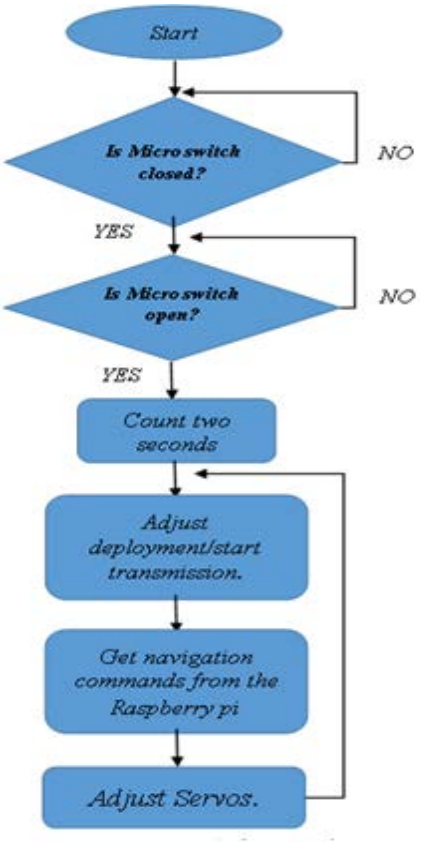

Figure 14. The flowchart of the Arduino steering system.

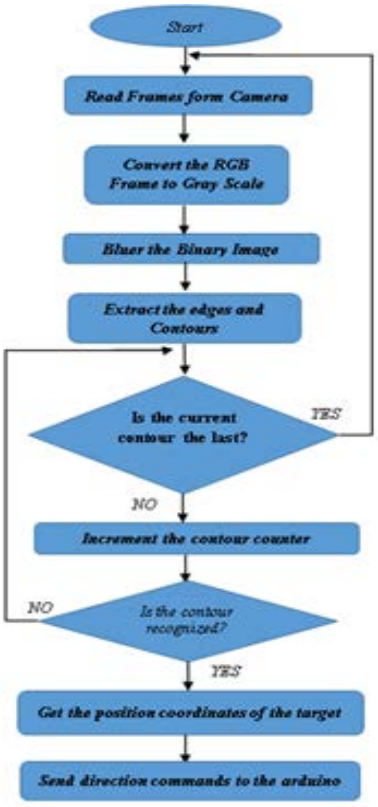

Figure 16. The flowchart for image processing and navigation system.

8. Conclusion

This paper presents a greenhouse gas measurement system. The system uses a cheap Quadcopter instead of expensive UAVs. The system measures the concentration of the GHGs at different altitudes. In addition, the system also generates data about temperature, humidity, and pressure. An on broad SD card records the data for later access. The system also transmits data to a ground station.

We implement two options for controlling the Quadcopter. The first option uses a PID algorithm. The second option uses the MultiWii configuration program. We integrate the system with a CanSat. This Cansat ensures safe landing of the system. To prevent damage during landing we use a specially designed parachute, which is controlled by an Arduino and two servomotors.

We test the system to measure greenhouse gases at different altitudes. The test results show that the system is able to measure greenhouse gases at different altitudes and environment. After completing the mission, the Quadcopter is able to land on a designated landing location. We develop an image processing based navigation and landing system. We use a Raspberry PI computer to execute the image processing algorithms.

In this work, we have measured greenhouse gas concentration in few selected locations. In order to monitor the environmental pollution level in the Arabian Gulf regions, we need to measure air quality parameters for an extended region in future. We use some commercially available sensors in our work. In order to make the system cost effective we need to replace these sensors with some low cost sensors. Recently, many innovations have been done in designing and manufacturing of low cost and reliable sensors. Some of these sensors can be found in the literatures [26-27]. In our future work, we will use these types of sensors in the present system.

REFERENCES

[1] John F. B. Mitchell, “The Greenhouse Effects and Climate Change”, Reviews of Geophysics, Vol.7, No.1, pp. 115-139, February 1989, https://doi.org/10.1029/RG027i001p00115

[2] Cécile Bessoui, Fabien Ferchaud, and Benoît Gabrielle, Bruno Mary, “Biofuels, greenhouse gases and climate change: A review”, Agronomy Sustainable Development, Vol. 31, No. 1, pp. 1-80, April 2011 https://doi.org/10.1051/agro/2009039

[3] United States Enviromental Protection Agency, “Climate Change and Ecosystem”, Retrieved from https://www.epa.gov/sites/production/files/signpost/cc.html accessed on June 22,2017

[4] National Oceanic and Atmospheric Administration, “Ecological Impacts of Climate Change”, Available http://oceanservice.noaa.gov/education/

[5] US National Climate Assessment, “Climate Change Impacts in United States”, Retrieved from https://pdfs.semanticscholar.org/b97c/5c098f9182886d0a4631108251d2ae0790f9.pdf

accessed on January 15, 2018

[6] New York Times Environment, Retrived from http://www.nytimes.com/2013/10/10/science/earth/by-2047-coldest-years-will-be-warmer-than-hottest-in-past.html , accessed on January 15, 2018

[7] Jane A. Leggett, "Federal Citations to the Social Cost of Greenhouse Gases", Report of Congressional Research Service, Available https://fas.org/sgp/crs/misc/R44657.pdf accessed on January 29, 2018

[8] John Moorehead and Time Nixon, "Global 500 Greenhouse Gases Performance 2010-2015:2016 Report on Trends", Available at https://www.thomsonreuters.com/content/dam/openweb/documents/pdf/corporate/Reports/global-500-greenhouse-gases-performance-2010-2015.pdf accessed on January 29, 2018

[9] Association for Unmanned Vechicle Systems International, "The Benefits of Unmanned Aircraft Systems: Saving Time, Saving Money, Saving Lives", Available https://epic.org/events/UAS-Uses-Saving-Time-Saving-Money-Saving-Lives.pdf accessed on January 29, 2018

[10] Lucas Satterlee, "Climate Drones: A New Tool for Oil and Gas Air Emission Monitoring", Environmental Law Institute, Washington DC, http://www.eli.org/sites/default/files/elr/featuredarticles/46.11069.pdf accessed on January 29, 2018

[11] E. Raymond Hunt, Craig S. T. Daughtry, Charles S. Walthall, James E. McMurtrey, Wayne P. Dulaney, "Agricultural Remote Sensing using Radio Controlled Model Aircraft", Digital Imaging and Spectral Techniques: Applications to Precision Agriculture and Crop Physiology, ASA-CSSA-SSSA, ASA Publication No. 66, pp. 197–205, 2003

[12] András Molnár1, "A Multi-rotor System for the Collection and Analysis of Measurements to Evaluate and Spatially Demonstrate the Pollutants in the Air", International Journal of Advanced Robotic Systems, Vol. 12, pp. 1-8, January 2015, https://doi.org/10.5772/61229

[13] J. J. Acevedo, B. C. Arrue, J. M. Diaz-Banez, I. Ventura, I. Maza, A. Ollero, "Decentralized strategy to ensure information propagation in area monitoring missions with a team of UAVs under limited communications", in Proc. International Conference on Unmanned Aircraft Systems (ICUAS), 28-31 Atlanta, GA, USA, pp. 565–574, May 2013, http:/doi.org/10.1109/ICUAS.2013.6564734

[14] D. Kingston, R. W. Beard, R. Holt, "Decentralized perimeter surveillance using a team of UAVs", IEEE Transactions on Robotics, Vol. 24, No. 6, pp. 1394–1404, January 2008, https://doi.org/10.1109/TRO.2008.2007935

[15] I. Maza, F. Caballero, J. Capitan, J. R. Martinez-de-Dios, A. Ollero, "A distributed architecture for a robotic platform with aerial sensor transportation and self-deployment capabilities", Journal of Field Robotics, Vol. 28, No. 3, pp. 303–328 May 2011, https://www.doi.org/10.1002/rob.20383

[16] A. Viguria, I. Maza, A. Ollero, "Distributed service based cooperation in aerial/ground robot teams applied to fire detection and extinguishing missions", Advanced Robotics,

Vol. 24, No. 1–2, pp. 1-23, April 2012, https://doi.org/10.1163/016918609X12585524300339

[17] A. S. Laliberte, A. Rango, J. E. Herrick: “Unmanned Aerial Vehicles for rangeland mapping and monitoring: A comparison of two systems”, in Proc. of the American Society for Photogrammetry and Remote Sensing Annual Conference, May 7-11, Tampa, FL, 2007 https://jornada.nmsu.edu/files/bibliography/07-033.pdf accessed on January 16,2018

[18] O.Sidek, A.Abdullah, U.N.Za’bah, N.A.Amran, et. al, “Development of Prototype System for Monitoring and Computing Greenhouse Gases with Unmanned Aerial Vehicle (UAV) Deployment”, in Proc. of International Symposium on Technology Management and Emerging Technology, May 27-29, Bandung, Indonesia, pp. 98-101, 2014 https://doi.org/10.1109/ISTMET.2014.6936486

[19] Malaver Rojas, Jairo Alexander, Gonzalez, Luis Felipe, Motta, Nunzio, Villa, Tommaso Francesco, Etse, Victor Kwesi, & Puig, Eduard, “Design and flight testing of an integrated solar powered UAV and WSN for greenhouse gas monitoring emissions in agricultural farms”, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 28 Sep - 02 Oct, Congress Centre Hamburg, Hamburg, Germany, pp. 1-6,2015 https://doi.org/10.3390/s150204072

[20] Chris Hugenholtz, and Thomas Barchyn, “A Drone in Search of Methane”, Project Summary–February 2016, University of Calgary available at http://ventusgeo.com/wp-content/uploads/2016/04/Project_outline_no_watermark-1.pdf accessed on January 16, 2018

[21] Juan Jesús Roldán , Guillaume Joossen , David Sanz , Jaime del Cerro and Antonio Barrientos, “Mini-UAV Based Sensory System for Measuring Environmental Variables in Greenhouses”, Sensors, Vol. 15, No.2, pp. 3334-3350, February 2015, https://doi.org/10.3390/s150203334

[22] Controller Simplifies, Available at https://radhesh.wordpress.com/2008/05/11/pid-controller-simplified/ accessed on January 29, 2018

[23] MultiWii software, Available at: http://www.multiwii.com/

[24] Dr. Adrian Rosebrock, "Practical Python and OpenCV: An Introductory, Example Driven Guide to Image Processing and Computer Vision", The 2nd Edition, 2015

[25] O. Javed and M.S. Yilmaz, “Object Tracking: A survey”, ACM Journal of Computing Surveys, Vol. 38, No. 4, pp. 1-45, December 2006, https://doi.org/10.1145/1177352.1177355

[26] Parra, L., Ortuño, V., Sendra, S., and Lloret, J. , “Two new sensors based on the changes of the electromagnetic field to measure the water conductivity”, Proceedings of the Seventh International Conference on Sensor Technologies and Applications (SENSORCOMM 2013), Barcelona, Spain ,Vol. 2531, p. 266-272

[27] Sendra, S., Parra, L., Ortuño, V., and Lloret, J., “A low-cost turbidity sensor development”, Proceedings of the 7th International Conference on Sensor Technologies and Applications (SENSORCOMM’13) , Wellington, New Zealand, December 03-05, pp. 266-272