Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Information Technology

(Department of Electronics & Telecommunication)

Group No.:- A03

Project Phase –I, Report

on

"SWARMING OF DRONES"

(SPONSORED BY: DRDO)

(Domain: ROBOTICS & IoT)

Submitted By:

Roll No.	Division	Name of Student	E-mail	Contact Number
411037	A	RUSHIKESH GAMARE	rushikesh.gamare@viit.ac.in	7020260394
411038	A	KUNAL GANDHI	kunal.gandhi@viit.ac.in	8208859454
411062	A	AMEY KORE	Amey.kore@viit.ac.in	8668872050

BE E&TC

Of

Savitribai Phule Pune University
(Formerly University of Pune)

Under supervision of

MR. VISHAL AMBHORE

Year 2019 – 2020

CERTIFICATE

This is to certify that **seminar work** entitled " **SWARMING OF UAV’S”** carried out in the seventh semester by,

Roll No.	Division	Name of Student
411037	A	RUSHIKESH GAMARE
411038	A	KUNAL GANDHI
411062	A	AMEY KORE

in partial fulfillment for the award of **Bachelor of Engineering** degree in Electronics and Telecommunication Engineering from Vishwakarma Institute of Information Technology, Savitribai Phule Pune University, Pune during the academic year 2019-20.

Date:13 DEC,2019

Mr.Vishal Ambhore
Guide

Mrs. Pallavi Deshpande
Project Coordinator

Dr. S. V. Kulkarni
H.O.D., E&TC Engg.

ABSTRACT

A UAV, in technological terms, is an unmanned aircraft. UAV’s are more formally known as unmanned aerial vehicles (UAV’s) or unmanned aircraft systems (UAS’s). These single UAV’s are capable of performing simple tasks like video recording, surveillance over small area.etc

When multiple UAV’s are used it becomes easy to cover a large area surveillance by dividing the area between the UAV’s. This combination of multiple UAV’s which take their own decision by communicating their neighbors is Swarming.

Swarming of UAV’s need a few more hardware and software to implement the required task. The UAV’s need to land precisely on the target without any errors, for implementing this IR lock sensor is added to the UAV. The testing of these UAV’s is a most tedious task to tackle. The UAV’s get commands using the manual remote or from the ground station which will send the wave points in terms of latitude, longitude and height using the Transmitter connected. But the GPS which helps the UAV reach the destination has issues like low Satellite count, GPS glitches in areas near the Airports. Hence, it was risky to implement the codes directly on the UAV’s. So the codes were successfully implemented on the simulation software’s available.

Further for communication between the UAV’s there should be a reliable method. The best way to communicate the real time data reliably to multiple UAV’s is done using ROS.ROS uses TCPROS protocol for reliable communication. The communication is implemented using Publisher-Subscriber model.

These UAV’s when used for surveillance need to detect the object present. Whether it’s a human, animal or any other objects like land mines. For detecting these objects precisely with high probability is the most important task which will be done using YOLO. YOLO is a neural Network based training and object detection method. It already is trained to detect the common objects around us which includes Bus, Car, Cat, Dog, Cycle, etc.

ACKNOWLEDGEMENT

This work could not have been completed without the guidance and encouragement of many people. We would like to particularly acknowledge those below.
We pay our humble regards and gratitude to **Jaywant Kolhe (Scientist D)** and **Dhanjay Rajmane (Scientist B)** from **DRDO** for guiding us and giving moral support and timely boost.

We wish to express our special thanks to **Prof. Vishal Ambhore(Project Guide)**,and project evaluators who helped us a lot in the preparation of our seminar topic.

INDEX

SR. NO.	CONTENTS	PAGE NO.
1	INTRODUCTION	6
2	LITERATURE SURVEY	7
3	OBJECTIVES	8
4	BLOCK DIAGRAM AND DESCRIPTION	9
5	METHODOLOGY	10
6	HARDWARE AND SOFTWARE REQUIREMENT	11
7	RESULT ANALYSIS AND DISCUSSION	14
8	APPLICATIONS	20
9	SCOPE OF FUTUTRE WORK	21
10	IMPLEMENTATION PLAN /PROJECT PHASES	22
12	REFERENCES	23

1. Introduction

Swarming UAV’s can be used in three ways by military forces: to attack, defend, and provide support functions such as intelligence, surveillance, and reconnaissance. Swarming is advantageous for offensive missions because it can overwhelm enemy defenses with a large number of potential targets. In a swarming attack the UAV’s are dispersed, which makes it difficult and expensive for the adversary to defend itself. If 10 UAV’s attack a target simultaneously and 7 are shot down, 3 will still be able to complete their mission. Because individual UAV’s in a swarm do not need to survive at a high rate, they may be dramatically cheaper than stand-alone weapon systems, which are often extremely sophisticated and expensive. It is possible that even a large swarm of several dozen unmanned aircraft may be both more effective and less expensive than a single manned or unmanned aircraft.

A UAV, in technological terms, is an unmanned aircraft. UAV’s are more formally known as unmanned aerial vehicles (UAV’s) or unmanned aircraft systems (UAS’s). Essentially, a UAV is a flying robot that can be remotely controlled or fly autonomously through software-controlled flight plans in their embedded systems, working in conjunction with onboard sensors and GPS.

For all these application few key features need to be added to the UAV’s. These includes a camera interfacing for surveillance, Obstacle avoidance sensors, Precision landing of UAV’s, communication between UAV’s and the ground station, communication between N UAV’s, object detection using AI.

2. Literature Survey

Considered to be one of the most resourceful and multipurpose innovations of the century, UAV’s have managed to perforate numerous segments of the global economic sphere [3]. Having uses in fields ranging from filmmaking to farming, UAV’s have managed to capture a major share of the commercial, personal goods, and military service market [4].

From quick deliveries at rush hour to scanning an unreachable military base, UAV’s are proving to be extremely beneficial in places where man cannot reach or is unable to perform in a timely and efficient manner. Increasing work efficiency and productivity, decreasing workload and production costs, improving accuracy, refining service and customer relations, and resolving security issues on a vast scale are a few of the top uses UAV’s offer industries around the world [9].

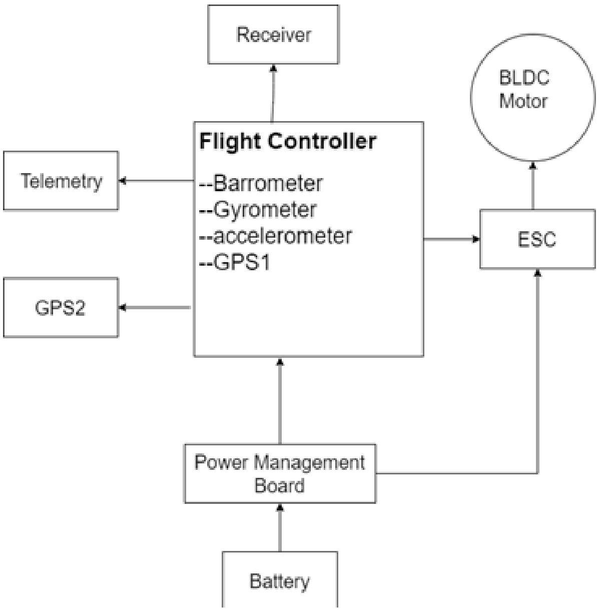
Military usage of UAV’s or RPAS (Remotely Piloted Aerial Systems) has become the primary use in today's world. Used as target decoys, for combat missions, research and development, and for supervision, UAV’s have been part and parcel of military forces worldwide [7].

UAV’s are used in situations where manned flight is considered too risky or difficult. They provide troops with a 24-hour "eye in the sky", seven days a week. Each aircraft can stay aloft for up to 17 hours at a time, loitering over an area and sending back real-time imagery of activities on the ground [2].

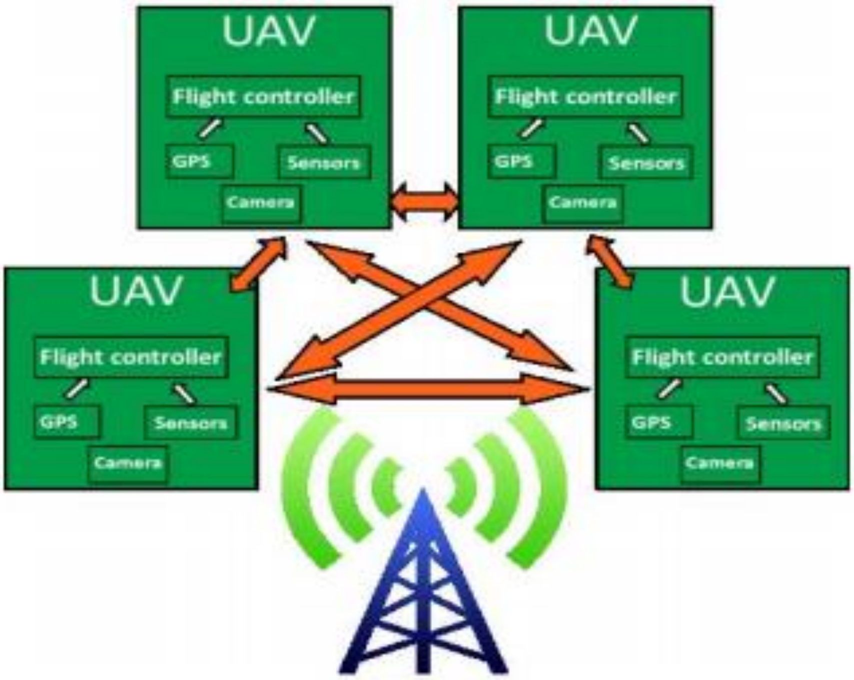
3. Objectives

- Precision landing.
- ROS Communication
- Swarm of UAV’s
- Video transferring from UAV’s to the ground station.
- Object detection using YOLO

4 Block Diagram



Basic UAV block diagram with all peripherals



→ Denotes communication through UDP or TCP

5. Methodology

Development of swarming needs building of small UAV’s capable of completing the assigned tasks. Moreover, its not just building of UAV’s, these UAV’s need to be safer from collision which is avoided by using obstacle avoidance using sensors.
Firstly, we assembled the 3 UAV’S & then selected the suitable firmware depending upon the stability. For this we used the ArduPiolet firmware then we tried to fly the UAV using QGroundControl software which provide user friendly environment. Then we set the mission which contain the GPS coordinate which were taken using map coordinator app. The UAV must follow the GPS coordinates.

Every UAV is connected with each other via WIFI (TP LINK) and every UAV contain ODROID Board which have ROS installed in it. which always transmit data over network and every UAV can communicate with each other & access each other’s data. Based on the data received by the UAV’s, the other UAV’s will take their own decisions.

Further testing of the UAV’s is a headache due to GPS errors. So, the simulation tools like Q ground controller which simulates the multiple UAV’s with the required task using a python script is used.

Component list

- Chassis
- BLDC motor
- Propeller
- Landing gears
- ESC
- Flight controller
- Power management board
- Battery
- GPS module
- Receiver
- Telemetry radio

6(a).Hardware description

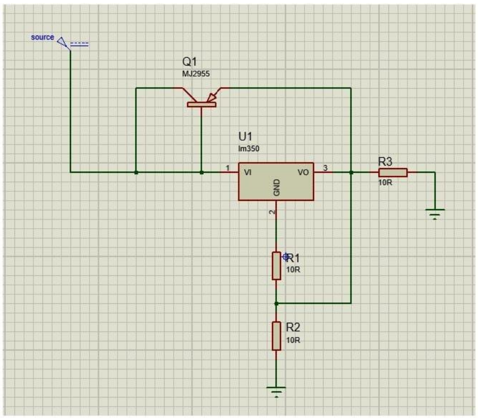
• **Chassis**	F450 chassis (selected) Weight- 282g F440 chassis Weight- 287g
• **BLDC motor**	1800kv rpm(selected) Max rpm at 12v – 21600 Weight – 48g No load current - 0.5A No load voltage – 10v Current capacity – 12A/60s Thrust generated with 3s battery and 1045 propeller – 1.2kg 2300kv rpm Max rpm at 12v – 27600 Weight – 40g No load current - 0.8A No load voltage – 12v Current capacity – 12A/60s Thrust generated with 3s battery and 1045 propeller – 1.5kg

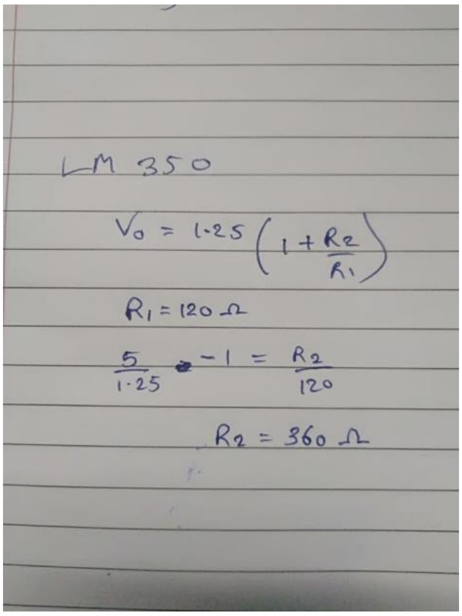
• **ESC**	20A hobbyking (selected) PWM frequency – 8k Hz to 16k Hz SIMNOK 30A ESC
• **Flight controller**	PIXHAWK4(selected) 2mb flash memory (ROM) 512 mb RAM Main soc microcontroller in px4 – STM32F427 Failsafe soc – STM32F100 APM 2.8 Flight controller 4mb flash memory (ROM) 512 mb RAM Main soc microcontroller in px4 – STM32F427 Failsafe soc – STM32F100
• **Power management board**	Input – 12v to 30v Output – 12v ,5v PWM
• **Battery**	3s LiPo 3000 MAh Min voltage- 10.2 V Nominal voltage – 11.1 V Max voltage – 12.6 V
• **Receiver – X8R**	16 channel duplex communication operating frequency - 2.4 GHZ Operating voltage – 4v to 10v Output voltage – 3.3 V Operating current – 100 mA at 5 V Max range – 1.5 km
• **Telemetry radio – 433Mhz**	Range – 3 km Operating voltage – 3 V to 6 V
• **Odroid microprocessor**	Odroid XU4

USB ports – 3
RAM – 2 GB
Input voltage – 5 V
Input current – 4A
8 GB eMMC5.0 HS400 flash storage
Operating system - Ubuntu 16.04

- **Power supply for odroid – 5v, 4A buck converter**

5v,4A lps for testing(using LM350),MJE2955





LM 350

$$V_o = 1.25\left(1 + \frac{R_2}{R_1}\right)$$

$R_1 = 120\,\Omega$

$$\frac{5}{1.25} - 1 = \frac{R_2}{120}$$

$R_2 = 360\,\Omega$

6(b),Required Software

- **Q ground controller**

QGroundControl provides full flight control and mission planning for any MAVLink enabled UAV. Its primary goal is ease of use for professional users and developers. All the codes are open-source source, so you can contribute and evolve it as you want.

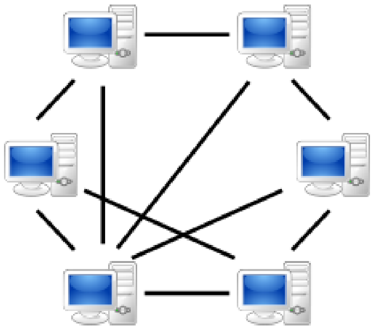
- **ArduPiolet**

ArduPilot is an open source, unmanned vehicle Autopilot Software Suite, capable of controlling autonomous: Multi UAV's, Ground rovers, Boats,Submarines.

MAVLink:

What is MAVlink?

- MAVLink is a very lightweight messaging protocol for communicating with UAV's (and between onboard UAV components).
- It follows a modern publish-subscribe and point-to-point design pattern.
- Publish/subscribe protocol for large size events wherein a centralized computing system is coupled with a decentralized BitTorrent-like peer-to-peer (P2P) protocol for scalable event distribution among publishers and subscribers.
- Peer-to-peer network shown below.



- Messages are defined within XML files.
  Which contains?

1. It has sender information.
2. It has receiver information.
3. It has a heading.
4. It has a message body.

Key Features:

- Very efficient. MAVLink 1 has just 8 bytes overhead per packet, including start sign and packet drop detection.

- MAVLink 2 has just 14 bytes of overhead (but is a much more secure and extensible protocol).
- MAVLink doesn't require any additional framing it is very well suited for applications with very limited communication bandwidth.
- Very reliable.
- It provides methods for detecting packet drops, corruption, and for packet authentication.
- Supports many programming languages, running on numerous microcontrollers/operating systems (including ARM7, ATMega, dsPic, STM32 and Windows, Linux, MacOS, Android and iOS).
- Allows up to 255 concurrent systems on the network (vehicles, ground stations, etc.)
- Enables both offboard and onboard communications (e.g. between a GCS and UAV, and between UAV autopilot and MAVLink enabled UAV camera).

DroneKit and DroneKit – SITL(simulation in loop)

What is DroneKit?

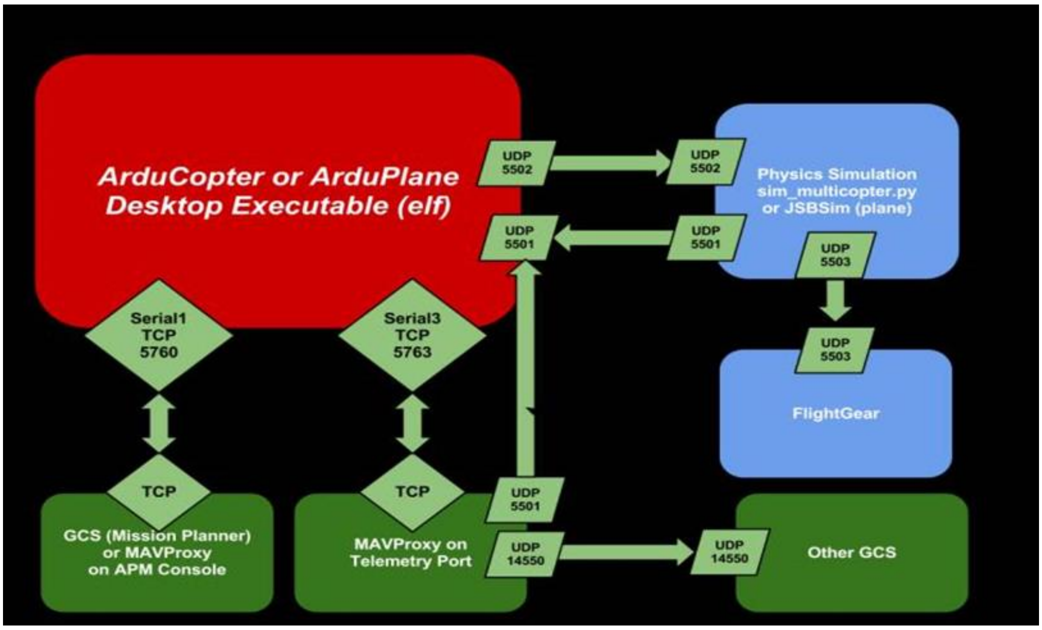
- Allows developers to create apps that run on an onboard companion computer and communicate with the ArduPilot flight controller using a low-latency link.
- Also used for ground station apps, communicating with vehicles over a higher latency RF-link.
- The API communicates with vehicles over MAVLink. It provides programmatic access to a connected vehicle's telemetry, state and parameter information, and enables both mission management and direct control over vehicle movement and operations.
- DroneKit Python APIs features:

  The API provides classes and methods to:

  1. Connect to a vehicle (or multiple vehicles) from a script
  2. Get and set vehicle state/telemetry and parameter information.
  3. Receive asynchronous notification of state changes.
  4. Guide a UAV to specified position (GUIDED mode).
  5. Send arbitrary custom messages to control UAV movement and other hardware (GUIDED mode).
  6. Create and manage waypoint missions (AUTO mode).

- The SITL (software in the loop) simulator allows you to run Plane, Copter or Rover without any hardware.

- SITL Architecture:


ArduCopter or ArduPlane Desktop Executable (elf)
UDP 5502
UDP 5502
UDP 5501
UDP 5501
Physics Simulation sim_multicopter.py or JSBSim (plane)
UDP 5503
UDP 5503
FlightGear
Serial1 TCP 5760
Serial3 TCP 5763
TCP
TCP
GCS (Mission Planner) or MAVProxy on APM Console
MAVProxy on Telemetry Port
UDP 5501
UDP 14550
UDP 14550
Other GCS

ROS (Robot Operating System)

What is ROS?

- ROS, the Robot Operating System, is an open source framework for getting robots to do things.
- The Robot Operating System (ROS) is a framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms.

Why should you learn/use ROS?

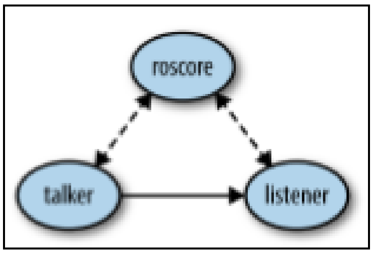
- The short answer is because it will save you time. ROS provides all the parts of a robot software system that you would otherwise have to write. It allows you to focus on the parts of the system that you care about, without worrying about the parts that you don't care about.

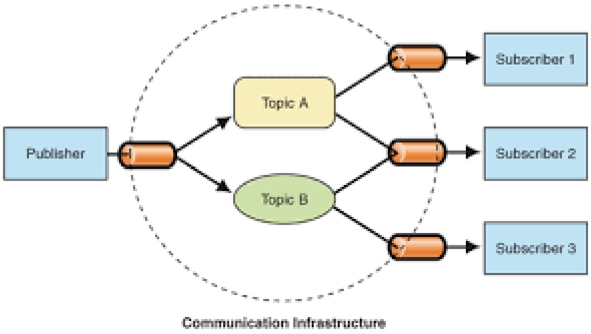
Roscore

- roscore is a service that provides connection information to nodes so that they can transmit messages to one another. Every node connects to roscore at startup to register details of the message streams it publishes and the streams to which it wishes to subscribe.

Node in ROS

- "Node" is the ROS term for an executable that is connected to the ROS network. Here we'll create the publisher ("talker") node which will continually broadcast a message.





PUBLISHER NODE

CODE(EXAMPLE)

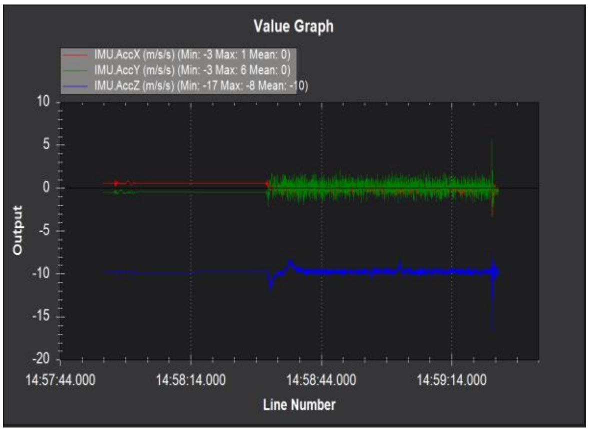
```
#!/usr/bin/env python
# license removed for brevity
import rospy
from std_msgs.msg import String

def talker():
pub = rospy.Publisher('chatter', String, queue_size=10)
rospy.init_node('talker', anonymous=True)
rate = rospy.Rate(10) # 10hz
while not rospy.is_shutdown():
hello_str = "hello world %s" % rospy.get_time()
rospy.loginfo(hello_str)
pub.publish(hello_str)
rate.sleep()

if __name__ == '__main__':
try:
talker()
except rospy.ROSInterruptException:
pass
```

SUBSCRIBER NODE

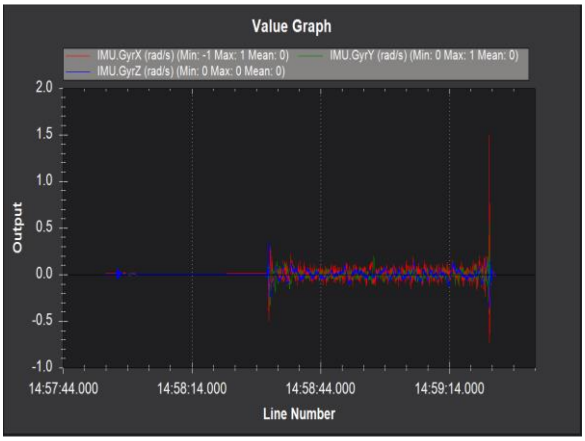
CODE(EXAMPLE)

```
#!/usr/bin/env python
import rospy
from std_msgs.msg import String
def callback(data):
    rospy.loginfo(rospy.get_caller_id() + "I heard %s", data.data)

def listener():

 # In ROS, nodes are uniquely named. If two nodes with the same
    # name are launched, the previous one is kicked off. The
    # anonymous=True flag means that rospy will choose a unique
    # name for our 'listener' node so that multiple listeners can
    # run simultaneously.
    rospy.init_node('listener', anonymous=True)
    rospy.Subscriber("chatter", String, callback)
    # spin() simply keeps python from exiting until this node is stopped
    rospy.spin()

if __name__ == '__main__':
    listener()
```

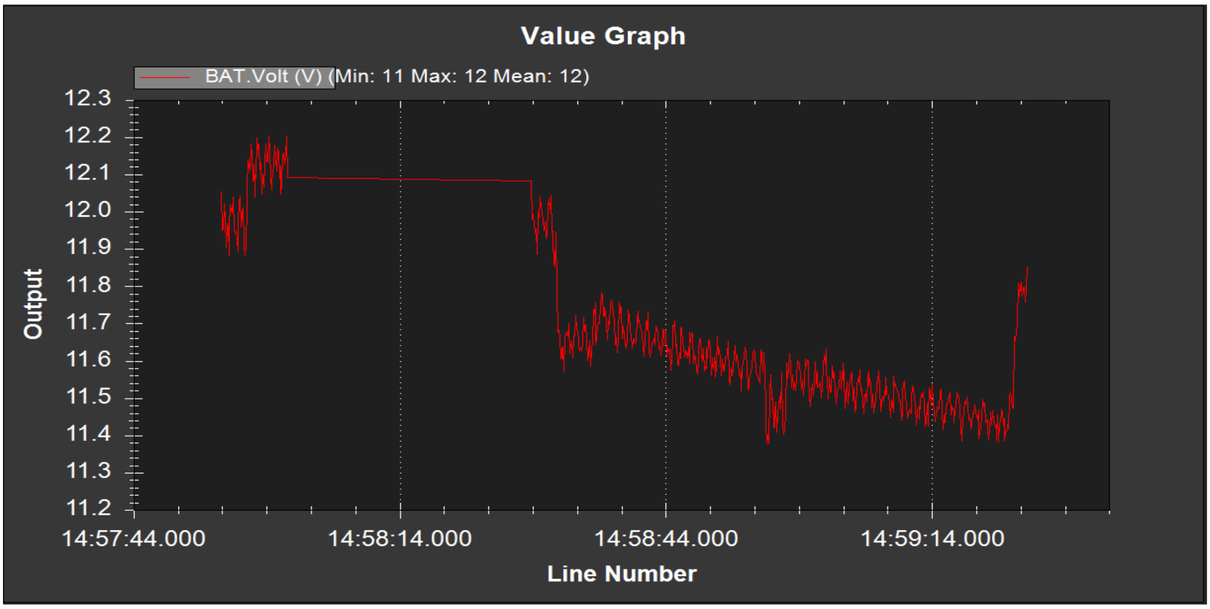
7.Result Analysis And Discussion



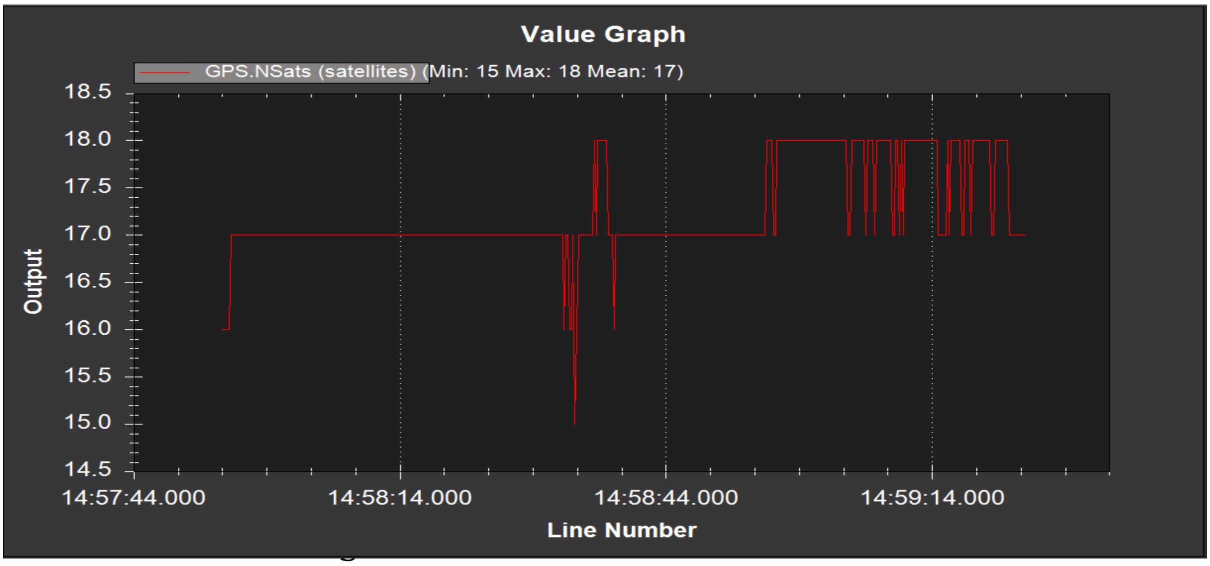
Accelerometer vs time



Gyroscope vs time



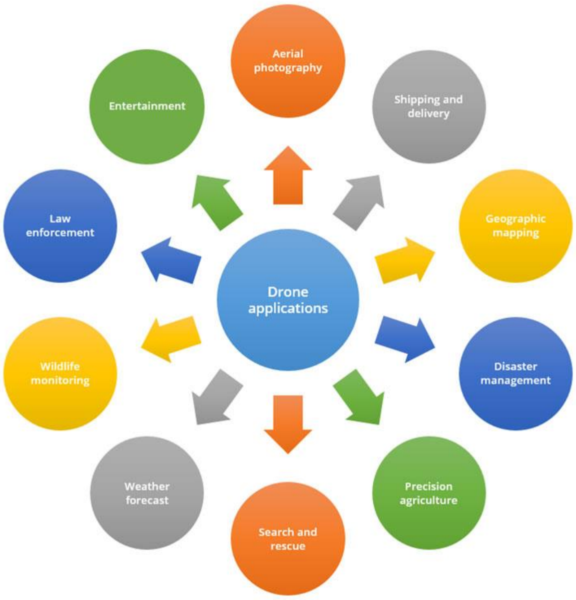
Voltage vs time



Satellite Count

8. Applications

- This type of UAV we can use for surveillance in prohibited area. This system is totally autonomous and reliable.
- We also use these UAV where it's not easy for humans to go easily.eg Windmills, Power Lines, etc.
- These UAV with object detection can track the enemy and send its location to the ground station.
- Advantages to swarm include time-savings, reduction in man-hours, reduction in labor, and a reduction in other associated operational expenses. The use of a coordinated number of sUAS surveying an entire farmstead with little to no operator intervention would greatly increase efficiency and could revolutionize precision agriculture.
- The most notable application of UAV swarm is delivery services. Amazon and United Postal Service have indicated interest in using UAS for package delivery (Amazon 2017; MacFarland 2017). Using a typical remote pilot and a single sUAS, package delivery would be inefficient. Swarms of UAV's with coordinated control and communication capabilities would be efficient in this application.
- Search and rescue, forest fire monitoring and fighting, flood and earthquake response, etc.
- First Aid delivery
- Military observation of regions
- Theatrical Performance
- Building inspections - more UAV's, more coverage
- Agriculture



9.Scope of Future work

- The data received by the robots need to be stored as a log and processed as per the assigned tasks.
- This data will be used for efficient implementation of swarming.
- For collision avoidance sensors need to be added on the four sides of the UAV.
- For object detection a camera and a processing unit is needed on each UAV. This requires a rigorous training on the huge data sets for efficient results.

References

1. J. Daugman et al. , “High confidence visual recognition of persons by a test of statistical independence”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1,148-1,161,1993.
2. "Information Technology, Biometric performance testing and reporting -- Part 1: Principles and framework, ISO/IEC 19795-1.,” 2006[12].
3. N. Poh and J. Kittler, "A Unified Framework for Biometric Expert Fusion Incorporating Quality Measures", IEEE Trans. on Pattern Analysis and Machine Intelligence, 34(1):pp 3-18, Year 2012.
4. http://wiki.ros.org/ROS/Tutorials[6]
5. https://www.theguardian.com/technology/2014/aug/15/harvard-kilobot-swarm-future-of-robotics
6. https://pdfs.semanticscholar.org/645e/6080cee6d31fdda903809692ea6ab58b5990.
7. https://www.researchgate.net/publication/264457775_Swarm_Intelligence_Concepts_Models_and_Applications[4]
8. https://www.hindawi.com/journals/isrn/2013/608164/[3]
9. http://www.techferry.com/articles/swarm-intelligence.html#aspects [1]
10. https://www.nrcresearchpress.com/journal/doi/10.1139/juvs-2018-0009?mobileUi=0#.XYLjUSgzbIV[5]
11. https://www.thebalancecareers.com/military-and-civilian-UAV-use-4121099
12. https://www.researchgate.net/publication/264457775_Swarm_Intelligence_Concept_Models_and_Applications [8]
13. https://www.hindawi.com/journals/isrn/2013/608164/[6]
14. https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/swarms[7]

Mr. Vishal Ambhore

Project Guide