GANTT Chart

Task	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8
Understanding components/Hardware collection								
Building/Algorithms for detecting failure								
Algorithm to stabilize								
Algorithms for moving the damaged drone with two propellers								
Testing and Refining								

GANTT Chart

Task	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8
Literature review/Hardware collection								
Building/Algorithms for detecting and stabilizing after failure								
Algorithms for moving the damaged drone with two propellers								
Testing and Refining								

SUMMER WEEKLY REPORT

Week 1-2

Ordered parts, worked on different softwares that may be useful – Matlab Simulink

We ordered the required hardware online except Pixhawk (which is the main component, a microcontroller), we didn't have it during our 1st 2 weeks. So, we used the Pixhawk that the aeromodelling club has for some days. But we couldn't get enough time to use the Pixhawk. So, we started working on Simulink – MATLAB that can simulate a drone in various situations, but we couldn't use it make/build custom codes for the drone in case of a failure. So, we had to work on Ardupilot software later in the 5-6 weeks which actually allows us to design custom control algorithms.

Week 3-4

Built the drone, Set up Raspberry Pi for communication from GCS(Ground Control Station-Computer) to Drone using WiFi. Literature study of Ardupilot Software.

In the 3rd week we got many of our project parts required except for the battery. We built the drone during this week and we setup raspberry pi which can be used as a companion computer on our drone. After building the drone we performed a preliminary communication test between the Pixhawk and Raspberry Pi so that we can proceed to study the source code. During 4th week we worked on understanding the architecture of the Ardupilot source code and studied different libraries and modules that we can use and edit to build our own customary failsafe.

Week 5-6

Made series of custom Ardupilot firmwares, started testing the custom firmwares and also changed the structural/mass distribution of the drone to make it stable during its decent and the uncontrolled spin due to the failure of the actuators.

We started customizing our failsafe into the source code on week 5. We worked on the landing mode using only two off the propellers and finished it. At first we had the problem of compiling the code due to issues with GCS (Ardupilot-Mission Planner), after figuring out we successfully compiled a firmware file with our landing mode and we also completed the preliminary communicating test of Raspberry Pi and Pixhawk. But we encountered a problem while doing the same communication with our custom firmware
instead of the one (Standard Ardupilot version) we used for our preliminary testing.

We then looked into this problem and figured out a way to establish a successful communication between Raspberry Pi and Pixhawk with our customary firmware. But later we realized that the Raspberry Pi will be redundant because the task at hand can be done just using a Pixhawk, which is a great thing because of the reduced complexity of the solution. Changed the structural/mass distribution of the drone to make it stable during its decent and the uncontrolled spin due to the failure of the actuators

We began testing out custom landing mode but due to some problems in the algorithm, during testing we had our drone frame damaged and we had to order the required parts on Amazon and rebuilt the drone after receiving the parts. So, we tested our flight mode, after some debugging. We achieved a flight mode which use only two motors to safely land in case of a motor failure but it kept declining at a higher rate. To find the reason behind we tested using throttle input through RC channel instead of the automated landing-throttle input.

Week 7-8

Re-iterated the proprietary fail safe firmware and also built detection algorithm.

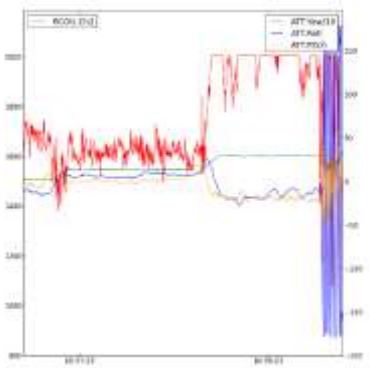
Finally settled on a firmware that helps increase the touch down time by 3 times and helped reduce the impact force considerably so that the damage is minimal.

From the results of 6th week we observed that with only two motors, the system cannot produce enough thrust to hover the copter even at the highest throttle input. We observed small acceleration in our flight mode during landing but our flight mode is working fine.

It was clear that motors with higher voltage rating, we will be able to produce a very steady descent using our flight mode. But, buying new and better motors will be expensive and at the same time, very time consuming because we the motors weren't available locally. So, we decide to test with a 2200 mAh battery instead of a 5000 mAh battery which helps decrease the thrust required to hover and during landing using only two motors. We conducted few tests with 2200 mAh battery where we obtained a steady descent unlike 5000 mAh battery. Which implies that our custom landing algorithm is working properly.

Now that we built the landing mode, we had to make detection algorithm that can detect the problem and thereby change the mode from Stable to Fail-Safe(Custom)

At first we thought of using gyroscope, accelerometer data to identify sudden ROLL and YAW development due the sudden change in thrust and torque that the failed motor produced prior to the failure. But due to inaccuracy of the sensors data in case of higher YAW rates and ROLL speeds the exactness of the data is compromised. So, we chose a different and better way to figure out the failure. In case of a motor failure (both internal motor and propeller failures) the thrust produced by the motor will be less than the output given by our Pixhawk, so in order to get the required thrust, the output given by Pixhawk to that particular motor will increase. But due to the failure either in motor performance or by a



propeller brakeage the output thrust will never reach the required thrust. So, the output given by Pixhawk of that particular failed motor increases to almost 100% while other motors are function at an output of 30 - 60 % in usual flights. In this case we can identify motor failure before it becomes catastrophic. If unusually high output is observed for an extended period (~40 cycles = 0.1 sec) (As shown in the figure) the failure is confirmed and we will activate our (Fail Safe mode) safe landing mode.

We created a new mode in which failure is induced to one motor by giving the output to that motor as a fraction of what it is supposed to be or even “0”. So that its Pixhawk output will behave the same in case of a motor/propeller failure. We then implemented both failure inducing algorithm and detection algorithm in a single firmware and tested it.

Result : The algorithm detected the failure and changed the mode automatically to our custom failsafe mode. We observed steady decline of height with 2200 mAh battery which suggests the completion of the solution for the problem statement.