Flipkart GRiD 2.0

“Autonomous Indoor Drone – Round 3”

Team Name : *Orbit*
Institute Name : *KIET Group of Institutions, Ghaziabad*

Team member details

Team Name	Orbit		
Institute Name	KIET Group of Institutions, Ghaziabad		
Team Members >	1 (Leader)	2	3
Name	Md. Aanis Noor	Pravesh Narayan Soni	Prateek Gupta
Batch	2022	2022	2022
Area of expertise	+Project Planning +Software testing/debugging +Computer Vision	+Embedded firmware +Hardware +Drone tech	+AI training +Software coding +Simulation
Achievements/ Participations/ Experience/ Works	+ Participated in Dronathon (by Uttarakhand Govt.) + Stood 2nd in DTMF controlled robot picking race by TechShiksha coaching institute. + Built a DTMF-call-only controlled UAV drone for college Technical Fest from scratch. + Worked/built 3+ UAV drones. + Worked on Arduino, linux, raspberry-pi.	+ Made 10+ motherboards (ARM) & some MEMS modules as well. + Participated in Dronathon (by Uttarakhand Govt.) + Worked on RTOS firmwares for ARM microcontrollers. + Made 10+ UAV drones. + Made own working flight controller from scratch and its software as well. + Conducted a drone race in college. Repository: https://github.com/Elvez?tab=repositories	+ Made smart glasses as an independent project + Worked on 'my parliament' app for Innovation Centre robot management + Participated in Smart India Hackathon

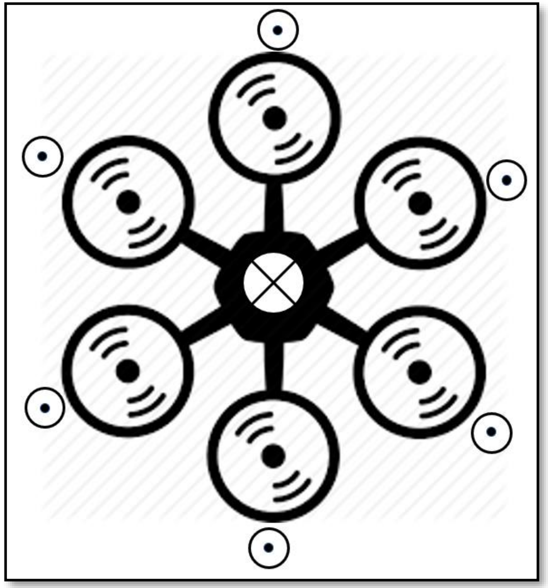
Drone specifications

Frame model / Material	Mini 290 (Carbon Fiber)
Frame wheelbase (mm)	290
Frame arm size (mm)	125 x 25 (L x W)
Multi-copter type	Hexacopter
ESC category	15A
Motor rating	2300kV BLDC
Propeller rating	6” 6030
Flight Controller	DJI Naza M-lite
Battery	3300mAh LiPO battery 3S
Claw servo	M-995 10kg Servo
Compass Module	HMC5883L
Master microcontroller	STM32F103C8
Camera for obstacle detection	Pixy2cam
IR Sensor for obstacle avoidance	Sharp GP2D12
RxTx	Radiolink R12DS pair
WiFi Tx for Camera	ESP826
Barometer / Altimeter	MS5611

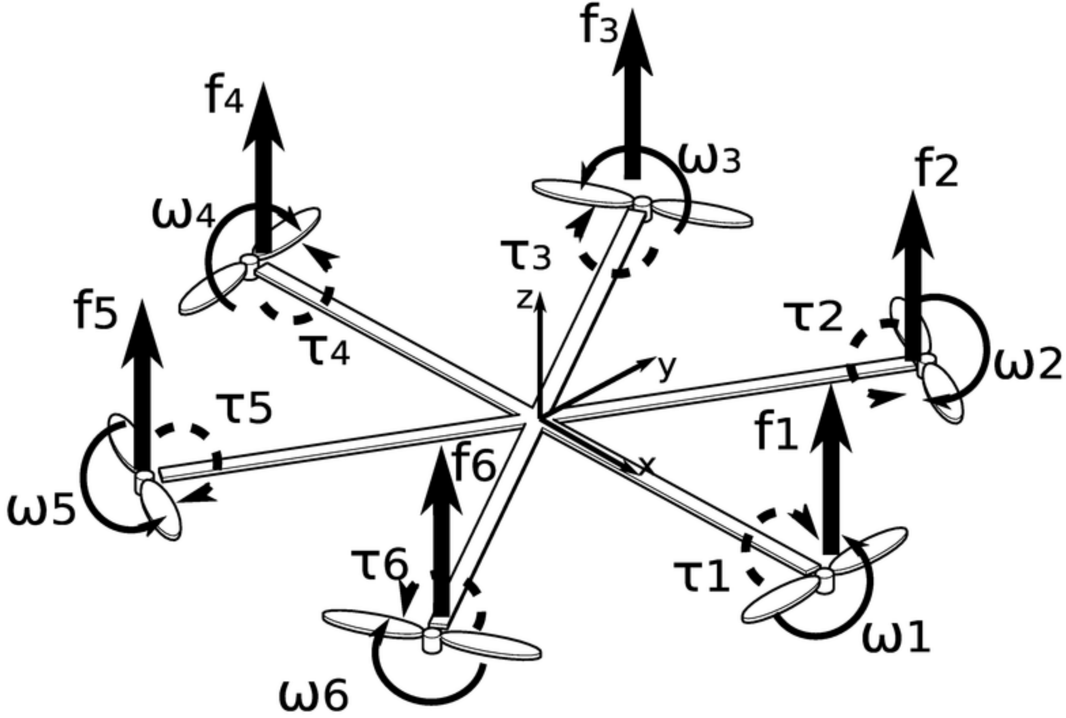
Payload Calculation

Thrust vector – max 1024gm for each motor

Payload vector – 6kg capacity (As centered as possible)



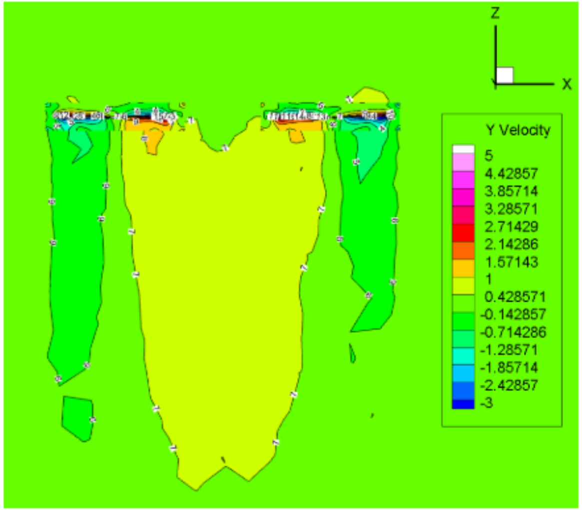
- Overall weight of drone with components (excluding payload) = 1,434gm
- Maximum thrust capacity left = 4,566gm
- By applying environmental filter for NTP and wind = 4,300gm
- Finally applying flying filter of (6.018:1) according to NTP and clear weather assumption = 2,300gm
- Hence for comfortable flying, the capacity of the drone is 2,300gm payload weight



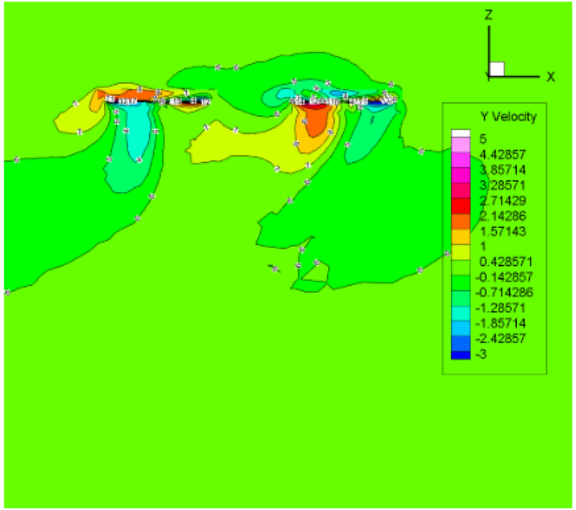
ω – Angular Momentum
τ – Torque
F – Thrust

Diagram : *Various mathematical components of the hexacopter UAV*

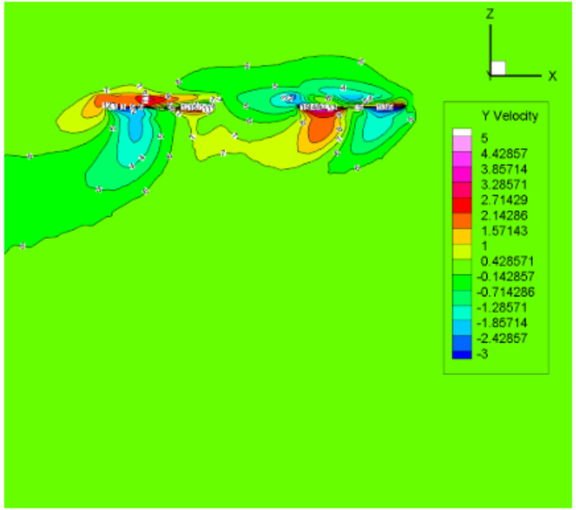
The velocity contour between the rotors with the influence of horizontal airflow is shown :



(a) 0 m/s

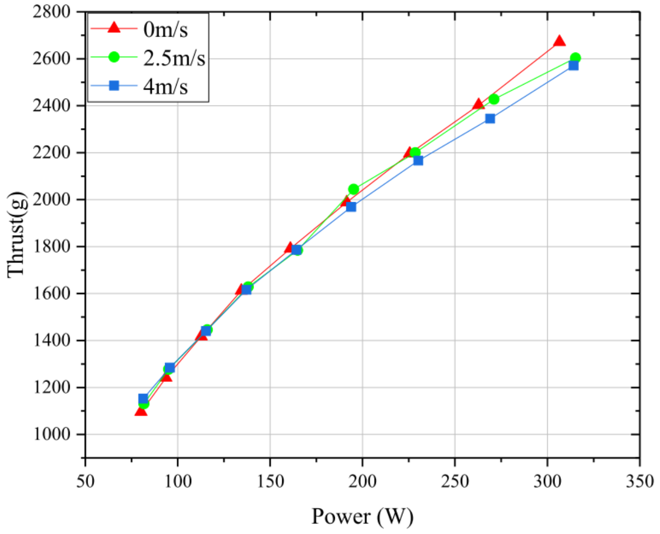


(b) 2.5 m/s

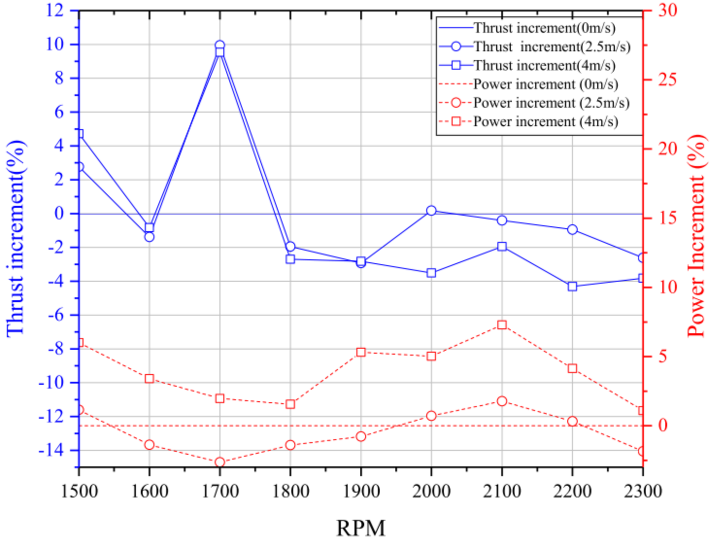


(c) 4 m/s

From an earlier made hexacopter, these were the results of experimental analysis using attached sensors :

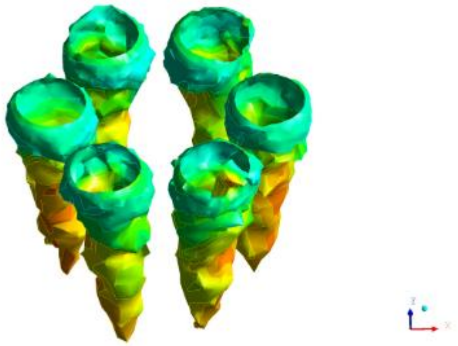


Graph(1): Thrust and power consumption with different wind speeds.

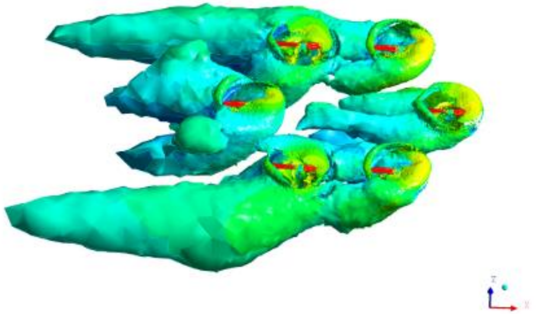


Graph(2): Thrust and power increment at different speeds.

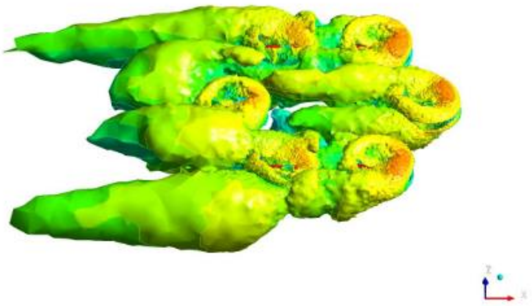
Vortex distribution of Hexacopter UAV at different horizontal speeds



(a) 0 m/s

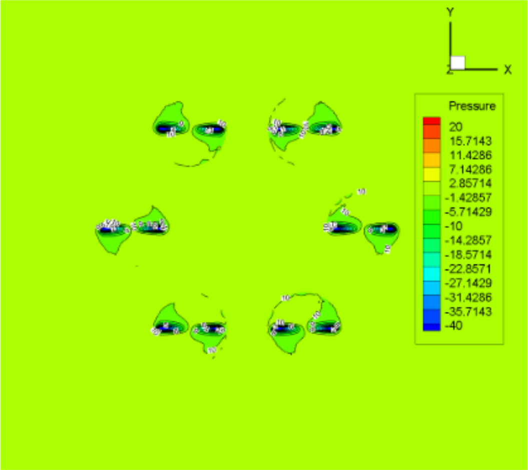


(b) 2.5 m/s

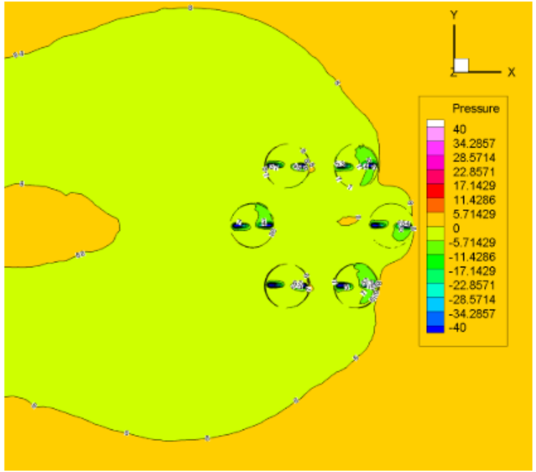


(c) 4 m/s

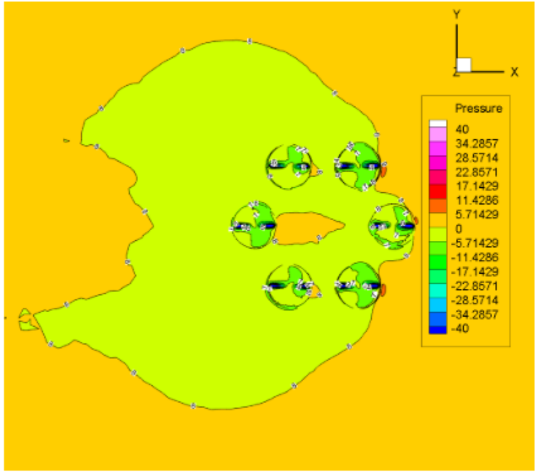
Pressure distribution on the plane upper 10 mm of the center of the Hexacopter UAV



(a) 0 m/s

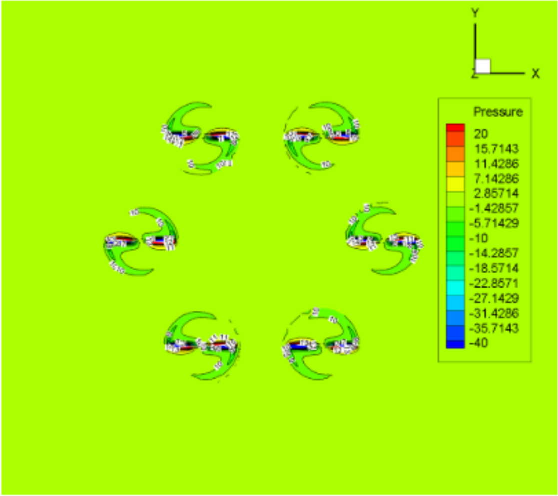


(b) 2.5 m/s

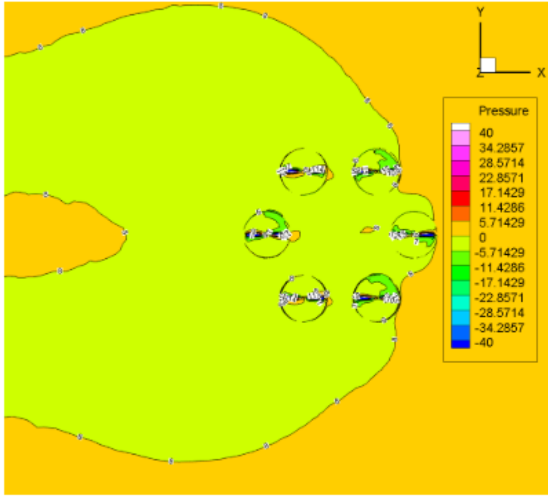


(c) 4 m/s

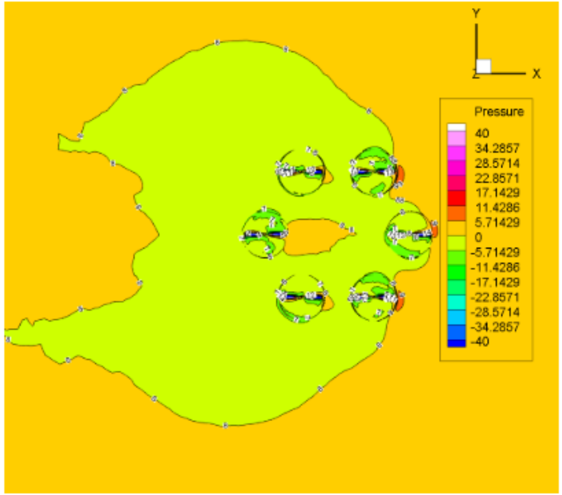
Pressure distribution on the center plane of the Hex-rotor UAV



(a) 0 m/s

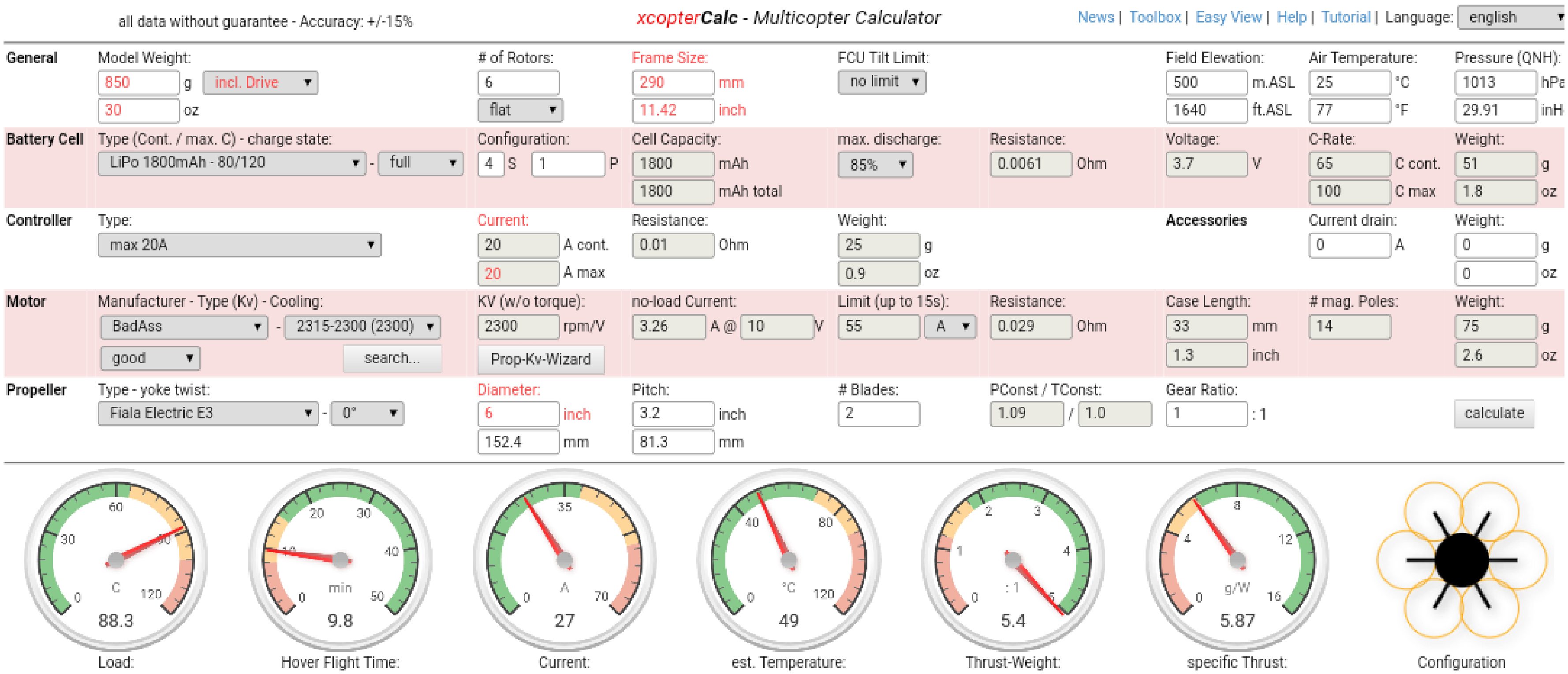


(b) 2.5 m/s



(c) 4 m/s

Performance Statistics



Computed using xCopter : https://www.ecalc.ch/xcoptercalc.php?ecalc&lang=en

(continued…)

Battery

Load:	88.33 C
Voltage:	11.81 V
Rated Voltage:	14.80 V
Energy:	26.64 Wh
Total Capacity:	1800 mAh
Used Capacity:	1530 mAh
min. Flight Time:	0.6 min
Mixed Flight Time:	4.2 min
Hover Flight Time:	9.8 min
Weight:	204 g
	7.2 oz

Motor @ Optimum Efficiency

Current:	28.36 A
Voltage:	11.25 V
Revolutions*:	23827 rpm
electric Power:	319.1 W
mech. Power:	258.3 W
Efficiency:	80.9 %

Motor @ Maximum

Current:	26.50 A
Voltage:	11.54 V
Revolutions*:	24630 rpm
electric Power:	305.9 W
mech. Power:	245.7 W
Power-Weight:	2159.3 W/kg
	979.4 W/lb
Efficiency:	80.3 %
est. Temperature:	49 °C
	120 °F

Wattmeter readings

Current:	159 A
Voltage:	11.81 V
Power:	1877.8 W

Motor @ Hover

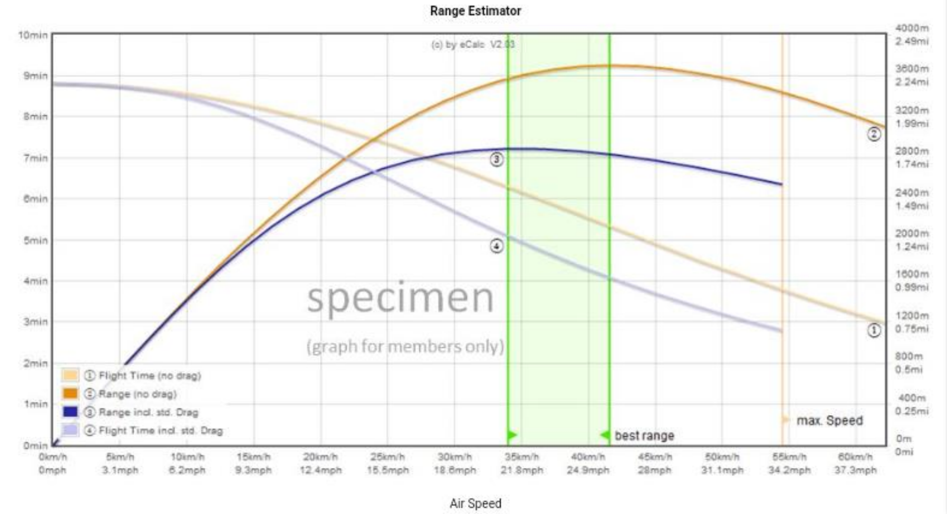
Current:	1.56 A
Voltage:	15.43 V
Revolutions*:	8800 rpm
Throttle (log):	17 %
Throttle (linear):	26 %
electric Power:	24.1 W
mech. Power:	15.9 W
Power-Weight:	173.1 W/kg
	78.5 W/lb
Efficiency:	66.1 %
est. Temperature:	28 °C
	82 °F
specific Thrust:	5.87 g/W
	0.21 oz/W

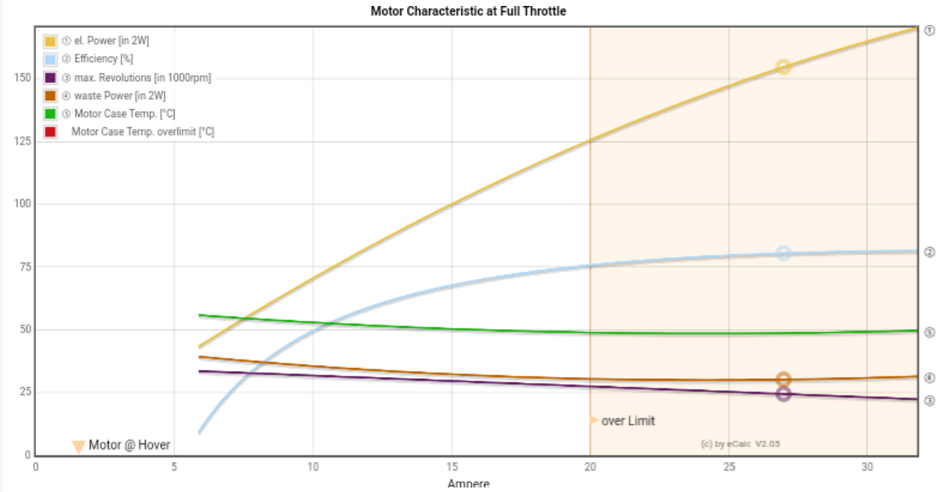
Total Drive

Drive Weight:	884 g
	31.2 oz
Thrust-Weight:	5.4 : 1
Current @ Hover:	9.38 A
P(in) @ Hover:	147.1 W
P(out) @ Hover:	95.6 W
Efficiency @ Hover:	65.0 %
Current @ max:	159.00 A
P(in) @ max:	2494.4 W
P(out) @ max:	1474.2 W
Efficiency @ max:	59.1 %

Multicopter

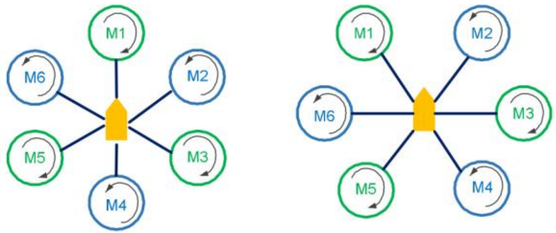
All-up Weight:	850 g
	30 oz
add. Payload:	3103 g
	109.5 oz
max Tilt:	78 °
max. Speed:	122 km/h
	75.8 mph
est. Range:	- m
	- mi
est. rate of climb:	22.6 m/s
	4449 ft/min
Total Disc Area:	10.94 dm²
	169.57 in²
with Rotor fail:	✓



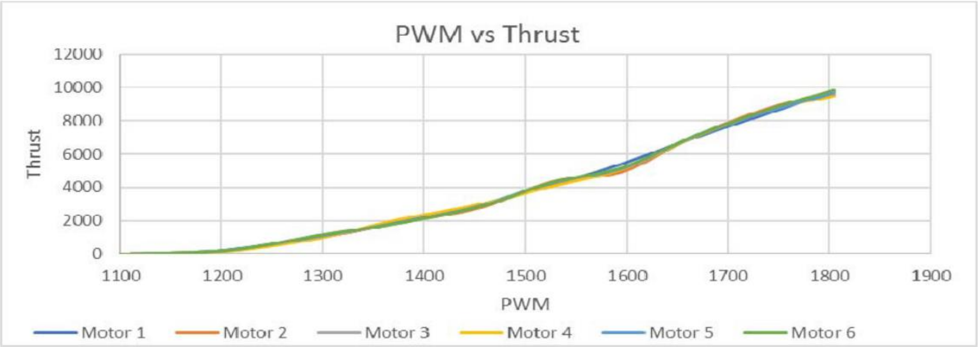


Structural & Stability Analysis (with & without payload)

Hexacopter configuration –

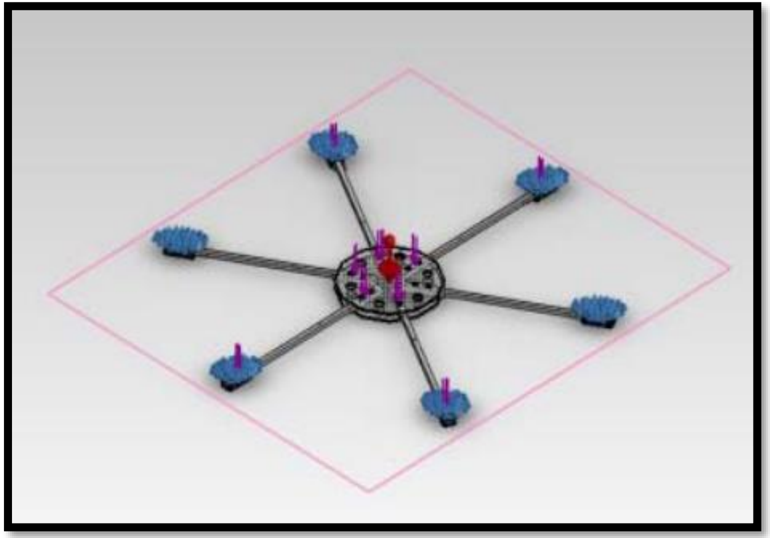


$$\text{thrust} = \frac{total\ weight\ X\ 2}{number\ of\ motors}$$

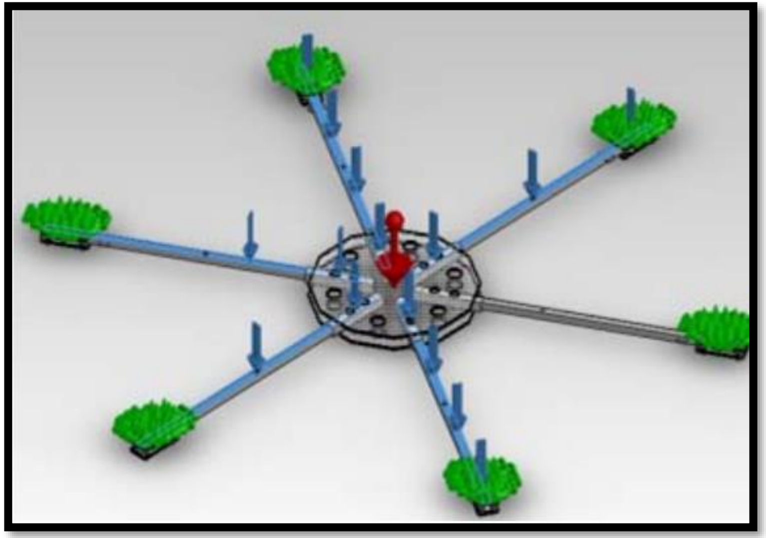


Note: We showcase a hexacopter made earlier with more realistic assumptions and simulation.

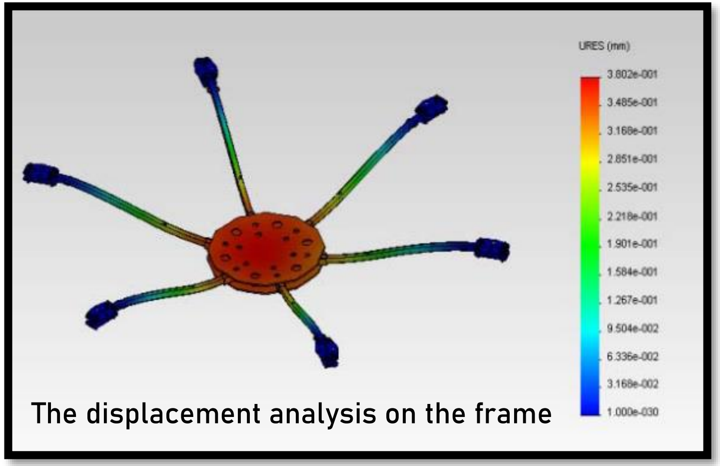
- *The Hexacopter frame consist of ABS material for its strength and lightweight characteristics.*
- *Furthermore, a **loading simulation** of this frame will be done by analyzing the static and dynamic materials. The force that is given as the load on this simulation is 50 N (more than required for better understanding of stress).*



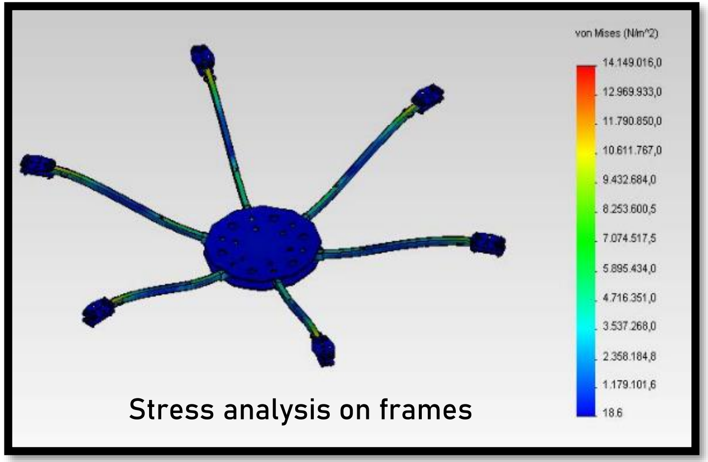
Hexacopter ABS frame design



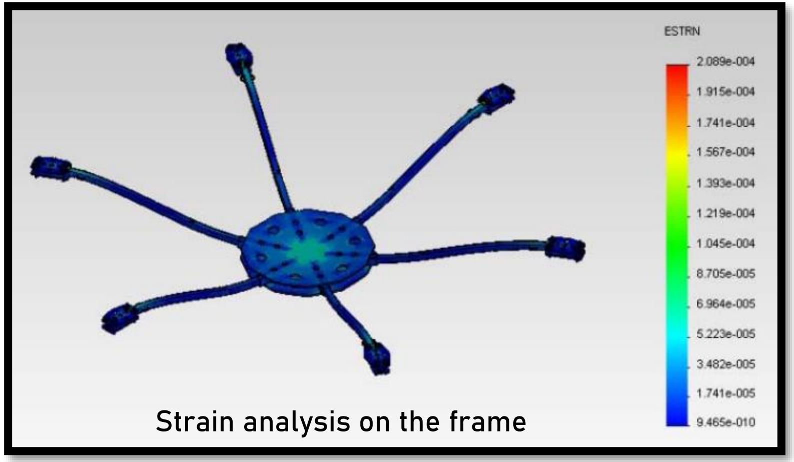
Placement of styles on the frame



The displacement analysis on the frame



Stress analysis on frames



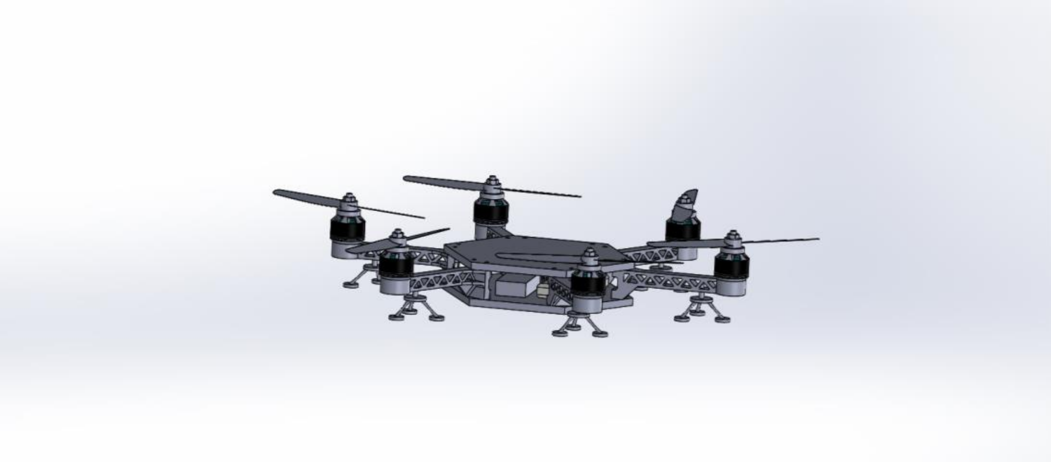
Strain analysis on the frame

CAD File with Payload & Drone Components

CAD File - https://drive.google.com/file/d/1TjKUJIND-tie4K79vyENH-ALdE3wVsSZ/view?usp=sharing

CAD File (gripper) - https://drive.google.com/file/d/1OpdvWDvvScgcSHJvw-XqJ5o8-uEJ5rBZ/view?usp=sharing

CAD zip (with part files)- https://drive.google.com/file/d/1H6qdqrtN59HDinxw74zl31VKuelBhMW4/view?usp=sharing



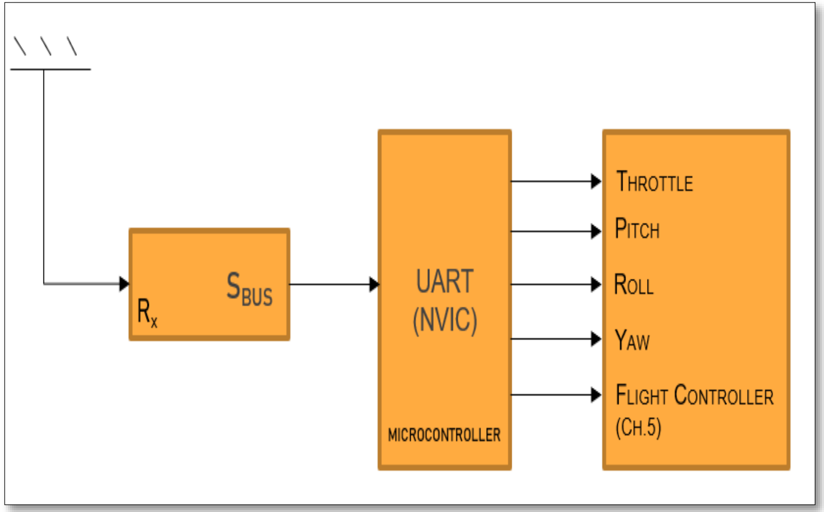
Hexacopter CAD file (snapshot)

Basic working principle:

A typical UAV drone is controlled mainly using the Flight Controller with manual PWM inputs.
But what we do is - Instead of connecting the receiver to the flight controller, we connect it to a micro-controller first and then use the global interrupt on the chip to drive the PWM connections on the flight controller by *TIM1_CH1*, channel 2, 3, 4 and *TIM2* (if we need).

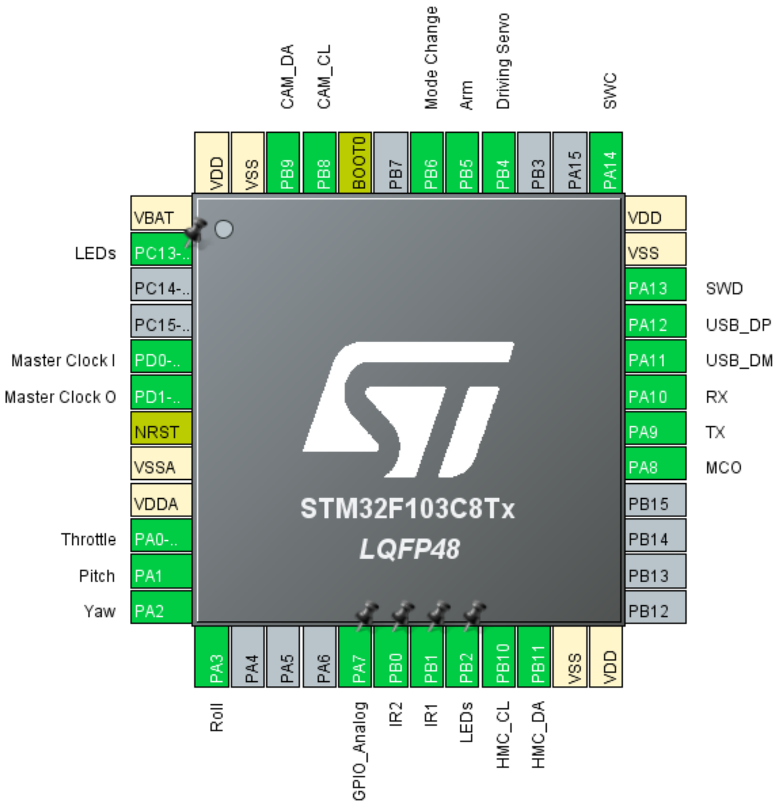
- '*TIM* is the TIMER peripheral of the microcontroller.
- We call this process "***Receiver Hijack Protocol (RHP)***"

Technically, instead of manually providing PWM via Tx, we embed the algorithm into the micro-controller (master) which then instructs the Flight Controller (slave) as dummy Rx/Tx.



(Refer to image on next page)

Video explaining RHP - https://drive.google.com/file/d/1nrVtzBPOb7A1DopLgrXx-qB41DZE8kKL/view?usp=sharing



Reference: *STM32 Microcontroller Pin Layout*

i) Take off and landing logic

Nutshell:

- The height for the given task is constant and pre-defined. We use an altimetric/barometric sensor to fix the height from the point of flight for take-off. Further to this, the flight and frame detection algorithms kick in.

- The final point of landing would be given a QR Code (printed image) which will be read by drone's AI-camera module; this determines the point of landing. The UAV will go stable approximately above this point, and the throttle will be lowered down until the UAV drone is on ground.

(continued...)

Take off Logic

Step (1)

- Fix Position on Land
- Power ON
- Arm Motors

Step (2)

- Throttle Up

Step (3)

- Wait for Altimeter Data
- if (altitude == definedHeight) :
  func(holdThisAltitude)

Landing Logic

Step (0)

- Associate a QR code to landing function
- Search for gates
- if (numberOfGates == None) :
  GoTo Step (1)

Step (1)

- Lurk & Look for a matching QR Code (through camera module)
- if (foundQrCode == 1) :
  GoTo Step (2)

Step (2)

- Open servo to drop item (after throttle down)
- Lower down throttle until UAV touches the ground.

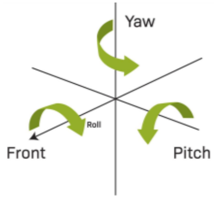
Extended : "Drop Mechanism & Landing"

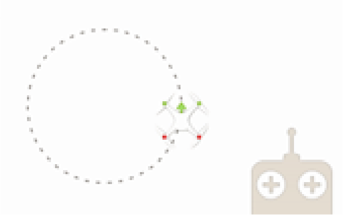
```
{
    repeat_process( );    //until signature = 0
}

//payload
{
    if (QrRead == wantedNumber)
        { writeServo(fullAngle);  //dropPayload }
}


{
    Land( );
    Disarm( );
}
```

ii) Flight algorithm

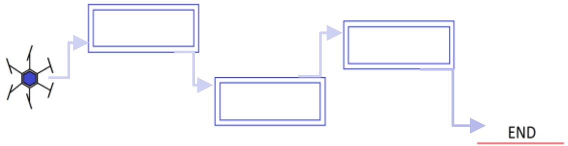
Nutshell:

- For the process of moving the drone automated, we first create a dataset by manually controlling it via PWM inputs of a transmitter; and then writing it into a file.
- We make functions for every movement of the drone via these experimental inputs.
- Finally, we apply the frame detection algorithm to automate drone movement in the environment.





RHP & Training



Use Transmitter to move drone in complex geometries → Write PWM data via interrupted UART (into a text file) → Apply frame detection logic to automate Drone movement as required

(continued…)

The dataset generation process:

- We use the transmitter to move the drone in complex geometries like turning, circling, quarter circling, etc.
- As the R_x gives data to the interrupted UART –

```
{
        HAL_UART_Receive_IT (&huart1, (uint16_t)pwm, 2, 10);
}

//&huart1 is a TypeDef to the HAL (Hardware Application Layer) peripheral of UART
//16-bit SBUS PWM data is received in "pwm', two bytes are received with a timeout of 10ms

{
        WriteToFile(pwm, File.txt);
}
```

- Through this process, we assume that we've generated these **four** basic functions (used for flight):

1. *moveForward(time); moveBackwards(time);*
2. *moveLeft(time); moveRight(time);*
3. *turnClockwise(angle);*
4. *turnAnticlockwise(angle);*

(continued...)

Memory Allocated: 5kB

Scheduler: Pre-emptive

Semaphores:
1 semaphore of 5 count

Number of tasks: 5

RTOS Block for Dataset Generation

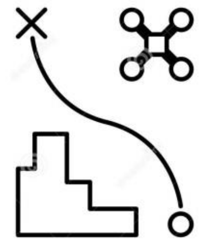
Dataset completion time: *3 days*

After we have the dataset, we proceed to make the code that allows us to move the drone as we simply like, according to the requirements.

iii) Trip management

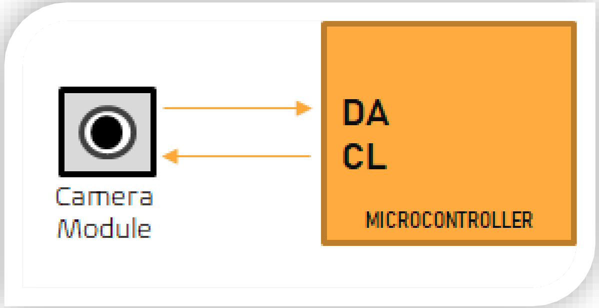
After successful take off, we go through the following steps for trip management :

1. Gathering data from the AI-Cam
2. Correction Algorithm
3. Find center of the corresponding gate & Alignment Algorithm
4. IR PID loop to avoid collision while passing through gates



(continued...)

A. *Gathering data from the AI-Cam*



* *Address* – 0xAE

The 16-bit ASCII return packet from pixy2cam contains all the info we need.

* *Packet* – { Rank ; Signature ; Vector X,Y ; Dimension(P_x) ; Age… }

* We only need the first 4 elements of the packet, so we move on to the correction algorithm…

B. Correction Algorithm

(PID loop 1)

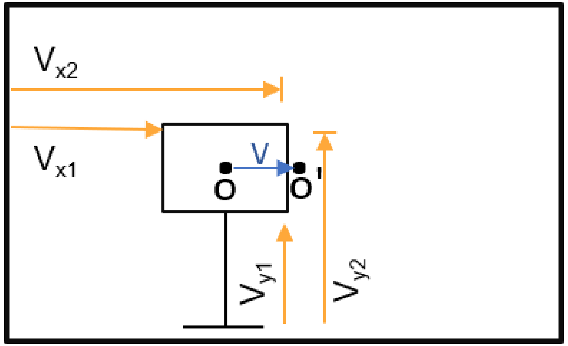
To extract only the first 4 elements (pseudo code) –

```
    do {
        camData[count] = camData[count] << 8 ;              //I2C data read and store
        camData[count+1] = camData[count+1] << 8 ;          //I2C data read and store
      }
  while {
        camData[count+3]
       }
```

The execution of above will leave us with only the first 4 elements which are required (from 8 elements)

Input data from the pixy2 AI camera module → Bit shift 8 bits (of 16 bits) to retain required data only → Apply logic to the acquired data for further processes

C. *Find center of the corresponding gate & alignment*



The vector X, Y read from I2C gives 4 vectors;

$L = | V_{x1} - V_{x2} |$

$B = | V_{y1} - V_{y2} |$

O => Center of Gate

O' => Center of Screen (known by default)

For O,

$O_x = [(V_{x2} - V_{x1}) / 2] + V_{x1}$;

$O_y = [(V_{y2} - V_{y1}) / 2] + V_{y1}$;

$O = (O_x , O_y)$

Now drawing a vector from O to O' through vector register (0xC20158) in pixy2

V => Vector between O and O'

Through the process below, the drone is centered according to the gate in front of it.

```
while (L:B == designatedRatio)
  {
    PID( );  //using vector V plotted earlier
  }
```

```
{
error = LVc ;
P = Kp * (Lv – Lc)  //Lv is needed length of vector (0)
                    //Lvc is the current length
while(Lvc < 3)      //very small error
  {
    I = KI * (Lvc) + I ;   //Integral
  }
    D = KD * (Lvc – Lvc1)   //currentError-previousError
PID = P+I+D ;  //actively uses move functions
```

All we need to do after this is

```
{  moveFowrward(time)  }          //time=> delayUntilSignature = Signature-1
                                  //signature getting reduced means there's one less gate in sight
```

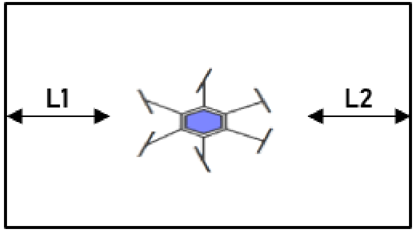
After this process, IR PID kicks in.

D. IR PID loop to avoid collision while passing through gates

While the drone goes inside the gates, the InfraRed sensor module gives data for PID correction of the drone to avoid any collision within the frame.

Interrupt of semaphore when drone is inside the gate :

The IR sensor reads along values…



We know that, $Length_{gate} = 1000mm$

$Height_{gate} = 500mm$

$Diameter_{drone} = 290mm$

So,

1000mm – 290mm = 710mm

Hence,

L1 = L2 = 710/2 = 355mm (for best results of centering inside gate)

(continued…)

IR PID loop:

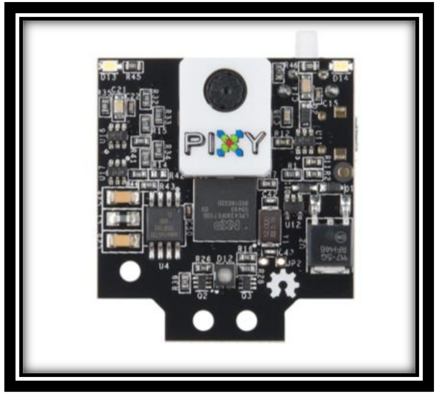
```
{

    P = K_p * (355 – L_1) ;
    while (L_1<3 || L_1<5)
        {
         I = (K_i * L_1) + I ;
        }
    D = K_D * (L_1 – L_1prev) ;    //current-previous error
    PID = P+I+D ;           //Actively uses more functions

}
```

Frame Detection Algorithm Details

By knowledge, L:B ratio is same for all gates & provided colour is '*RAL1023*'
This makes us not worry about other similar looking gates.

- *We grab image data as explained in Slide 17*
- *We centre the drone as explained in Slide 18/19*
- *We adjust drone inside frame using IR PID as explained in Slide 20/21*

All this was done after the frame was detected.
For frame detection, we use pixy2cam (camera) AI module available after-market.



pixy2cam module

Provided frame dimensions were fixed. In figures:

$$Length_{gate} = 1000mm$$
$$Height_{gate} = 500mm$$

- The Length:Height ratio is fixed as 2:1 (or 2.00 as float)
- Given frame colour - RAL1023

"We use this ratio and colour 'RAL1023' to train the AI camera module for detecting frames on-the-go using in-built operations."

<u>Rank</u> : Assigned number to multiple same objects in sight or off sight.

- Gates in sight are given ranks by the AI cam. We call them as signature.
- The drone will keep looking for gates unless no gate is in sight.

```
while(framesInSight != 0)
        { signature = signature-1 }
```

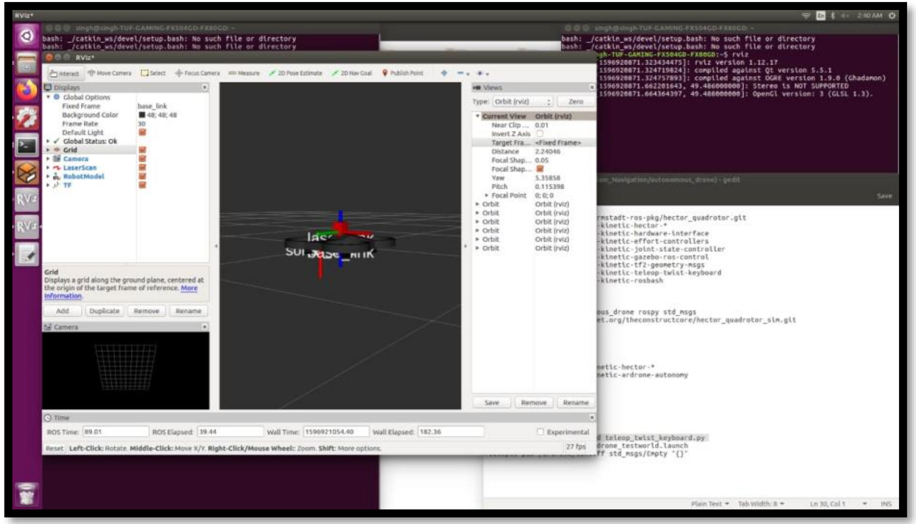
- When the frame are no more in sight and (signature == 0):

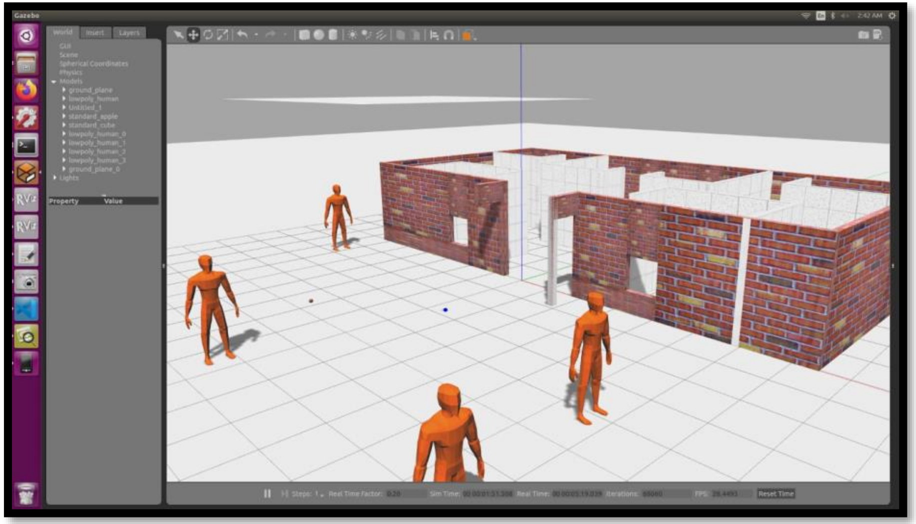
```
Execute(landing_procedure)
```

Reference: https://www.youtube.com/watch?v=EcCbEWiyiQY

Gazebo Simulation & Assumptions

Gazebo Simulation Resources : https://drive.google.com/drive/folders/1cQmLsdZa3bAO2rOxepxtQAlwnBhihTJf?usp=sharing





We've provided the link to resources (in use and development).

Unfortunately for us, Gazebo simulation was incomplete due to unfortunate events with our teammate's family. We were not able to meet this requirement within the deadline.

Component Sources, Price & total cost

Compass/Magnetomemter HMC5883L	₹370/-	https://robu.in/product/gy-273-hmc5883l-3-axis-module-magnetic-field-sensor/
Barometer MS5611	₹900/-	https://robu.in/product/gy-63-ms5611-01ba03-high-precision-pressure-sensor-height-sensor-module/
IR Sensor x2 (sharp **GP2Y0A41SK0F**)	₹1200/- (600x2)	https://robu.in/product/sharp-ir-distance-measuring-sensor-unit-4-30-cm-cable/
Motors 2300kv x6	₹4500/- (750x6)	https://robu.in/product/rs2205-2300kv-cw-brushless-motor-fpv-racing-quad-motor-fpv-multicopter-qav250-qav300/
6" Propeller Pairs (CW+CCW) x 3	₹900/-	https://robu.in (or local market)
1500 mAh LiPO battery Orane	₹2800/-	https://robu.in/product/orange-1500mah-4s-100c200c-lithium-polymer-battery-pack-lipo/
15A ESC x6	₹3000/- (500x6)	https://www.roboelements.com/product/30a-esc-brushless-bldc-motor-electronic-speed-controller-for-quadcopter/?gclid=Cj0KCQjwvb75BRD1ARIsAP6LcqsZredyqG3bi0cnp8qZr1oxryIL_Wqly5hLrXrpgqNPt85X168MTLkaAiiSEALw_wcB
Carbon Fibre Drone Printing	₹4000/-	https://robu.in
Pixy2 Camera	₹7300/-	https://robu.in/product/pixy-1-0-smart-vision-sensor-object-tracking-camera/
Master Microcontroller	₹500/-	https://robu.in/product/stm32f103c8t6-minimum-system-board-microcomputer-stm32-arm-core-board/
10kg Servo Motor	₹500/-	https://robu.in/product/towerpro-mg945-digital-high-speed-servo-motor-standard-quality/
flight controller	- Own -	-
Receiver and Transmitter	- Own -	-
WiFi Tx for camera	₹2000/-	https://robu.in
Reflective balls for motion capture training (x5)	- Own -	
TOTAL APPROXIMATE COST -	32-35k INR	

Component Details

Component ---	Specifications --
IR Sensor	3-80cm accuracy and range
AI Camera	640x480 @60Hz
Claw servo	10kg drive
Motors	2300kv, 0.000434kt
Frame Material	Carbon Fibre (Twill weave)
Flight Controller	8Ch Naza mLite
Radio	R12DS FHDSS 12 channel
Battery	1500mAh LiPO 4s 40C
ESC	15A with large decouple
Propellers	6" ABS 6030
Master MCU	ARM Cortex M3 72mHz

Execution plan with timelines

- Drone build time = 1-2 days
- Drone motion AI training and RHP training = 3-4 days
- PID tuning for IR correction = 3 days
- Gate detection training = 7 days
- PID tuning for gate detection and obstacle avoiding = 5 days
- Autonomous mission training and logging = 4 days
- GCS and GUI creation = 3 days

We need a total of <u>25-28 days</u> for full completion of the product, once we get a hands on for all the modules and parts for it.

Current Progress

- ✓ Minimal viable firmware ready
- ✓ Action plan ready
- ✓ Hardware assembly as soon as the college re-opens
- ✓ AI training logic ready

References

https://www.ecalc.ch/xcoptercalc.php?ecalc&lang=en

https://ieeexplore.ieee.org/document/7849648

https://www.researchgate.net/

https://www.udemy.com/course/robotics-with-ros-autonomous-drone-with-path-planning-slam/

https://www.semanticscholar.org/

https://www.simscale.com/

https://www.wired.com/2014/05/modeling-the-thrust-from-a-quadcopter/

https://www.instructables.com/id/Design-Build-and-Improve-a-Quadcopter/

https://airshaper.com/