SAFER DRONES

REAL-TIME OBSTACLE AVOIDANCE IN QUADCOPTERS

THE PROBLEM

Quadcopters are often deemed to be unsafe. Learning how to fly them is a challenge, and requires a great amount of focus.

Many recent autonomous quadcopters come with object detection (and avoidance) capabilities.

While flying a quadcopter manually, the pilot needs to always have the drone within sight. Even then low-altitude flight is difficult, due to more obstacles.

IDEAS

Quadcopter obstacle detection and avoidance during manual flight.

- Define a way to override pilot controls to take drone to safety.
- Minimum possible deviation from pilot's intended flight path.
- Notifying the pilot of obstacles around the drone and suggested flight path.

RESEARCH

Analyze different methods of autonomous control such as bang-bang, sliding scale. Sliding scale leaves pilot with some control to correct flight path.

Trade-offs between Ultrasonic sensors and a 3D camera. Implement foundation for a seamless and efficient reporting system using Kafka Streams.

Finding ways to merge pilot input and input from the avoidance unit and format it to be processed by flight controller.

RESULTS

Basic sliding scale avoidance system built.

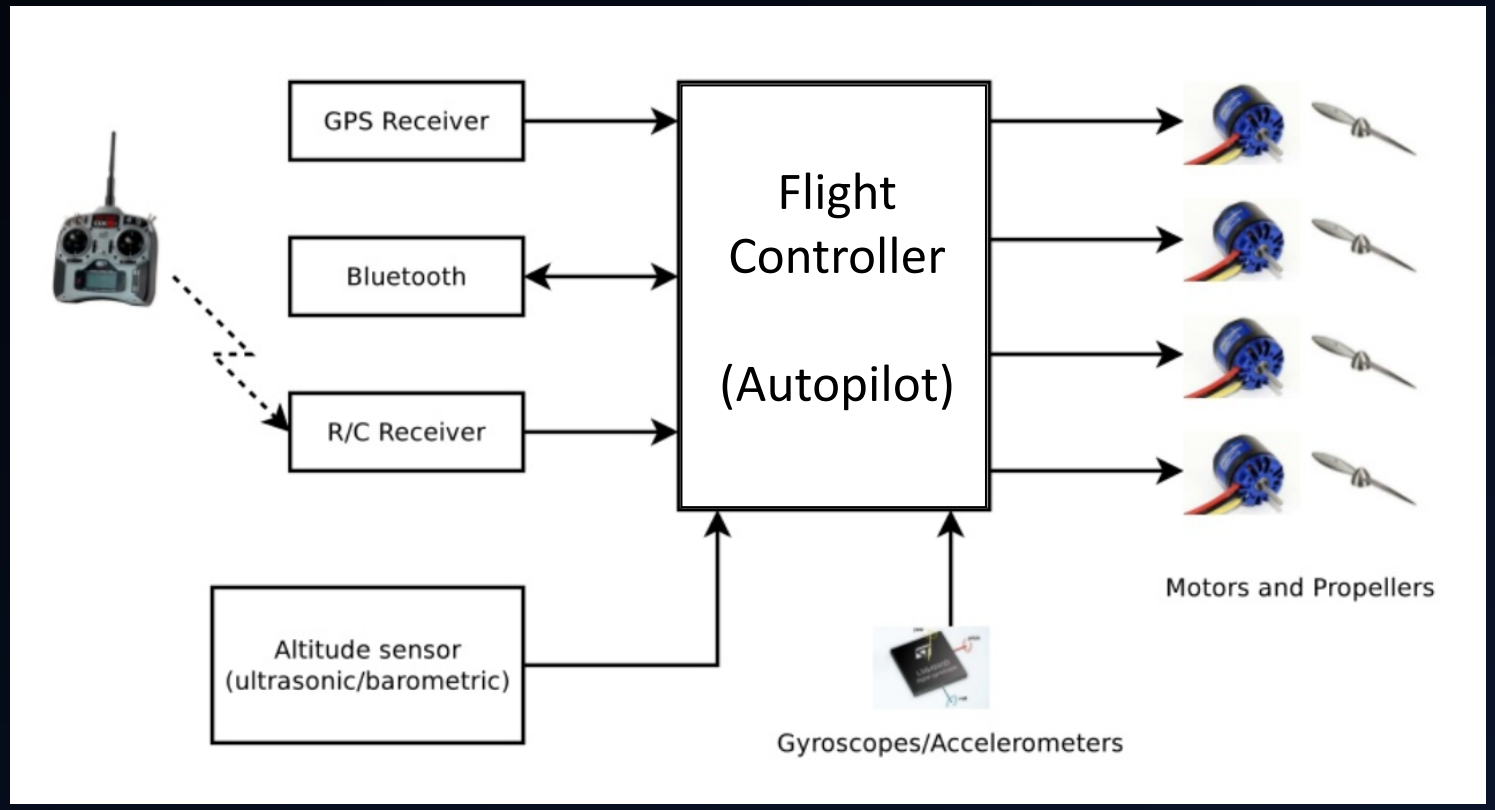
Custom built drones are hard to get up and flying! Onboard IMUs require extensive calibration.

Using ultrasonic sensors effectively a better alternative than a 3D Camera.

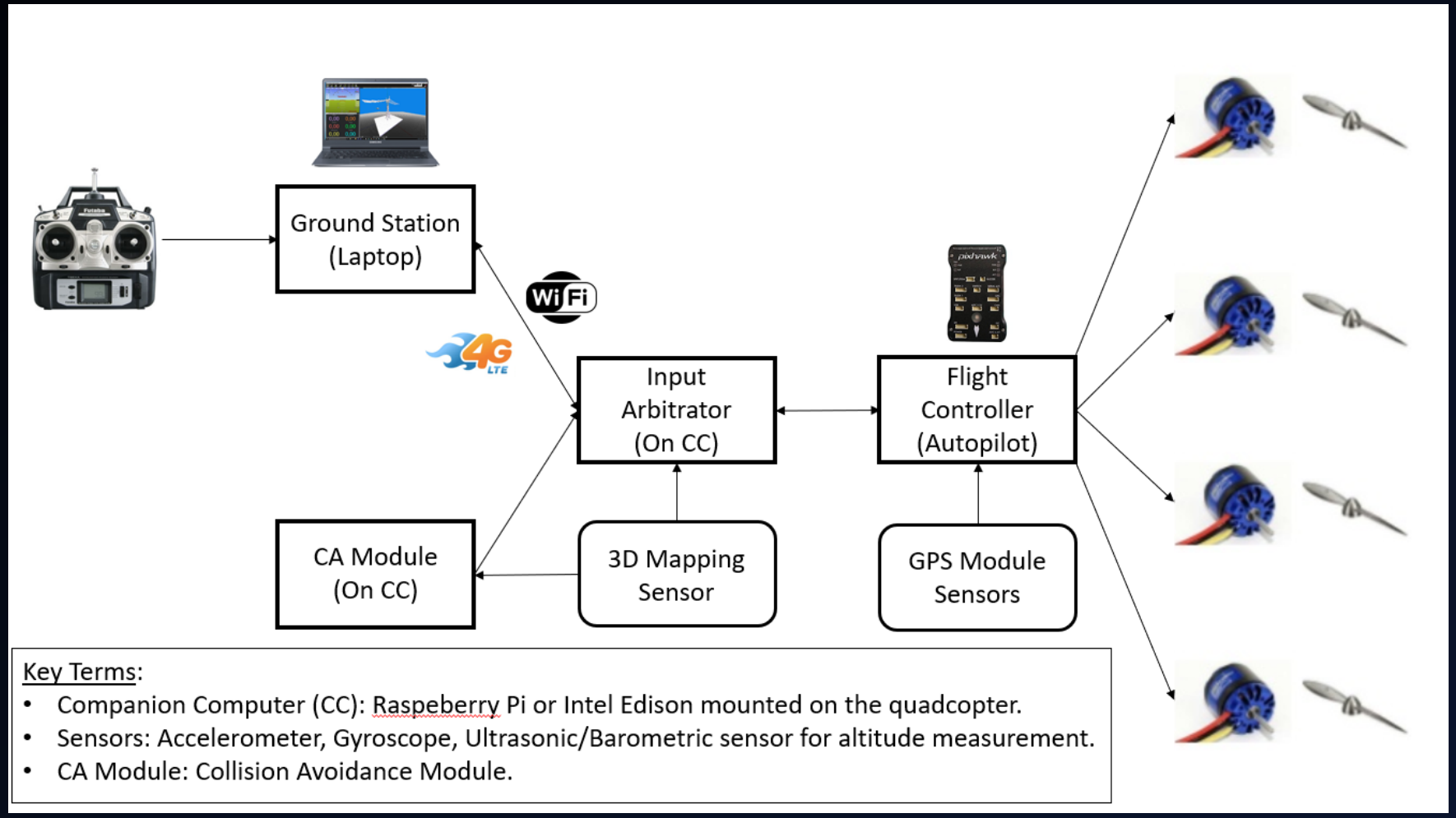
Future work

Integration of pilot and avoidance signal by unpacking MAVProxy packet and editing velocity values.

Background


GPS Receiver
Bluetooth
R/C Receiver
Altitude sensor
(ultrasonic/barometric)
Flight
Controller
(Autopilot)
Motors and Propellers
Gyroscopes/Accelerometers

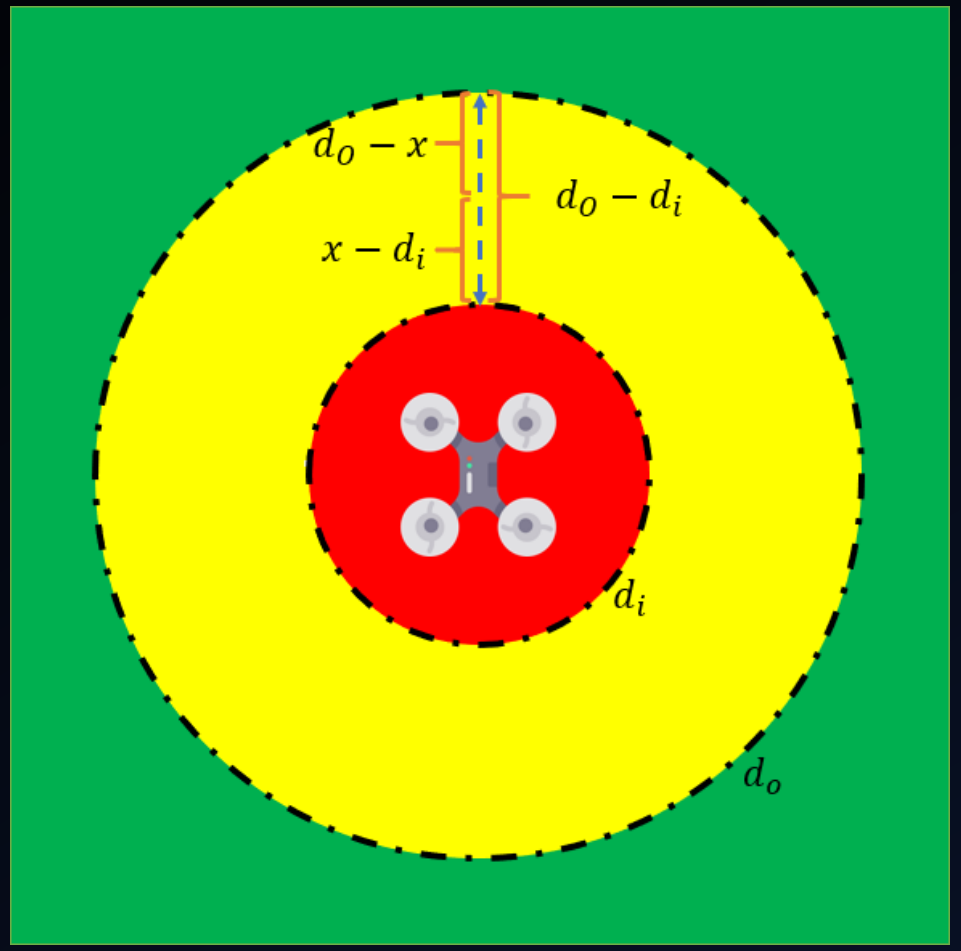
Idea



Key Terms:

- Companion Computer (CC): Raspeberry Pi or Intel Edison mounted on the quadcopter.
- Sensors: Accelerometer, Gyroscope, Ultrasonic/Barometric sensor for altitude measurement.
- CA Module: Collision Avoidance Module.

Idea



$$V^P = (v_x, v_y, v_z)^P$$
$$V^C = (v_x, v_y, v_z)^P$$
$$V^R = (v_x, v_y, v_z)^R$$
$$\propto * V^P + (1 - \propto) * V^c = V^R$$
$$\propto = \frac{x - d_i}{d_o - d_i}$$

V^P: *Velocity inputs from pilot*

V^C: *Velocity inputs from companion computer*

V^R: *Resultant velocity inputs*

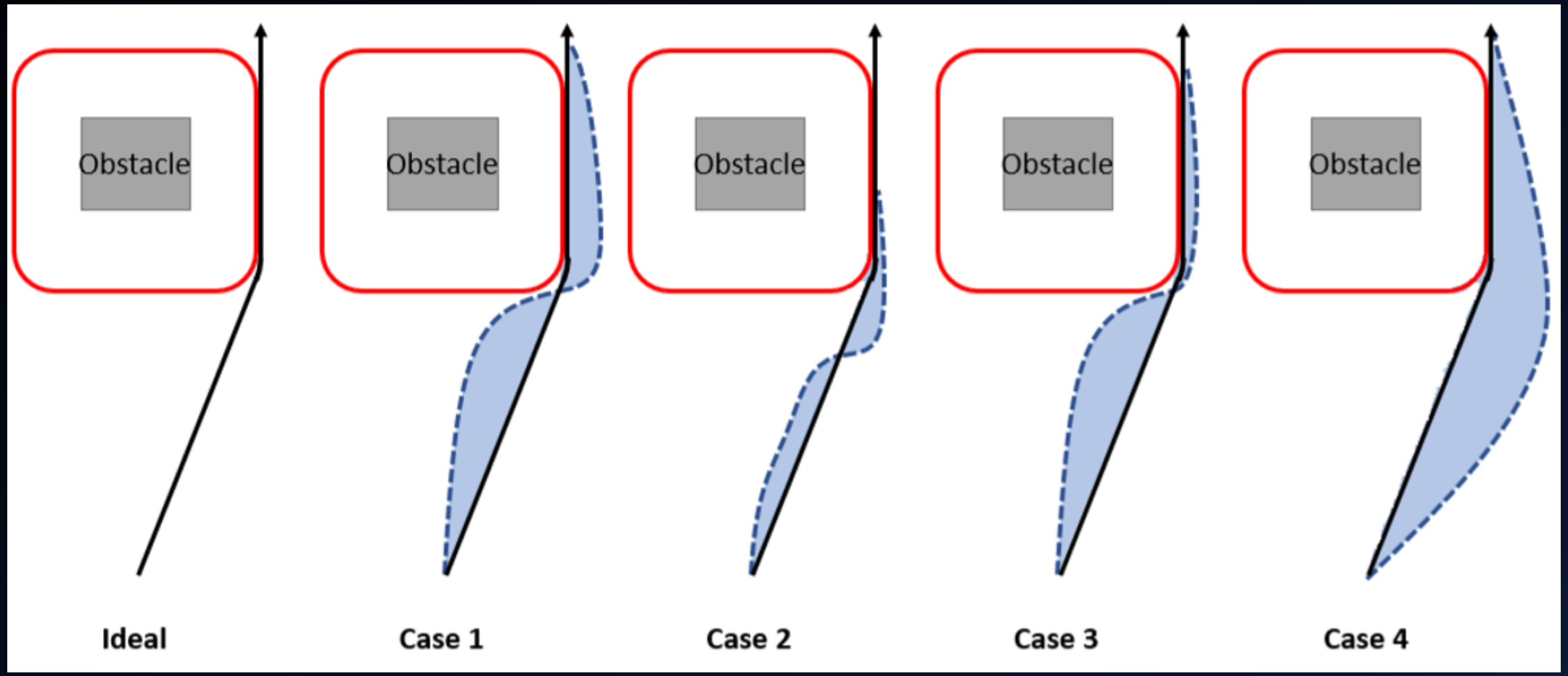
d_o: *outer radius of control zone*

d_i: *inner radius of control zone*

$\propto$: *control ratio for pilot*

x: *distance of closest point of obstacle that is obstructing flight path*

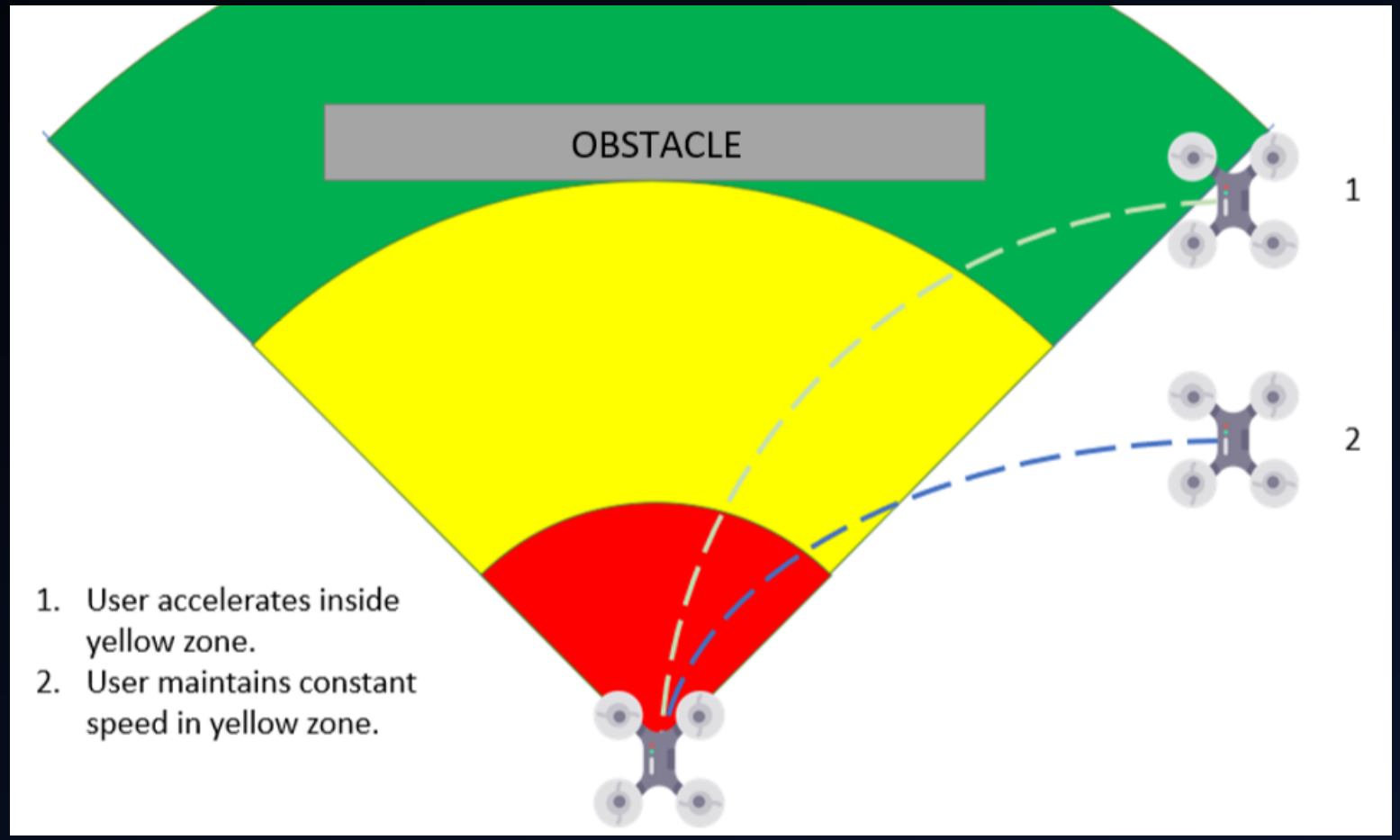
Potential Applications – Pilot Training



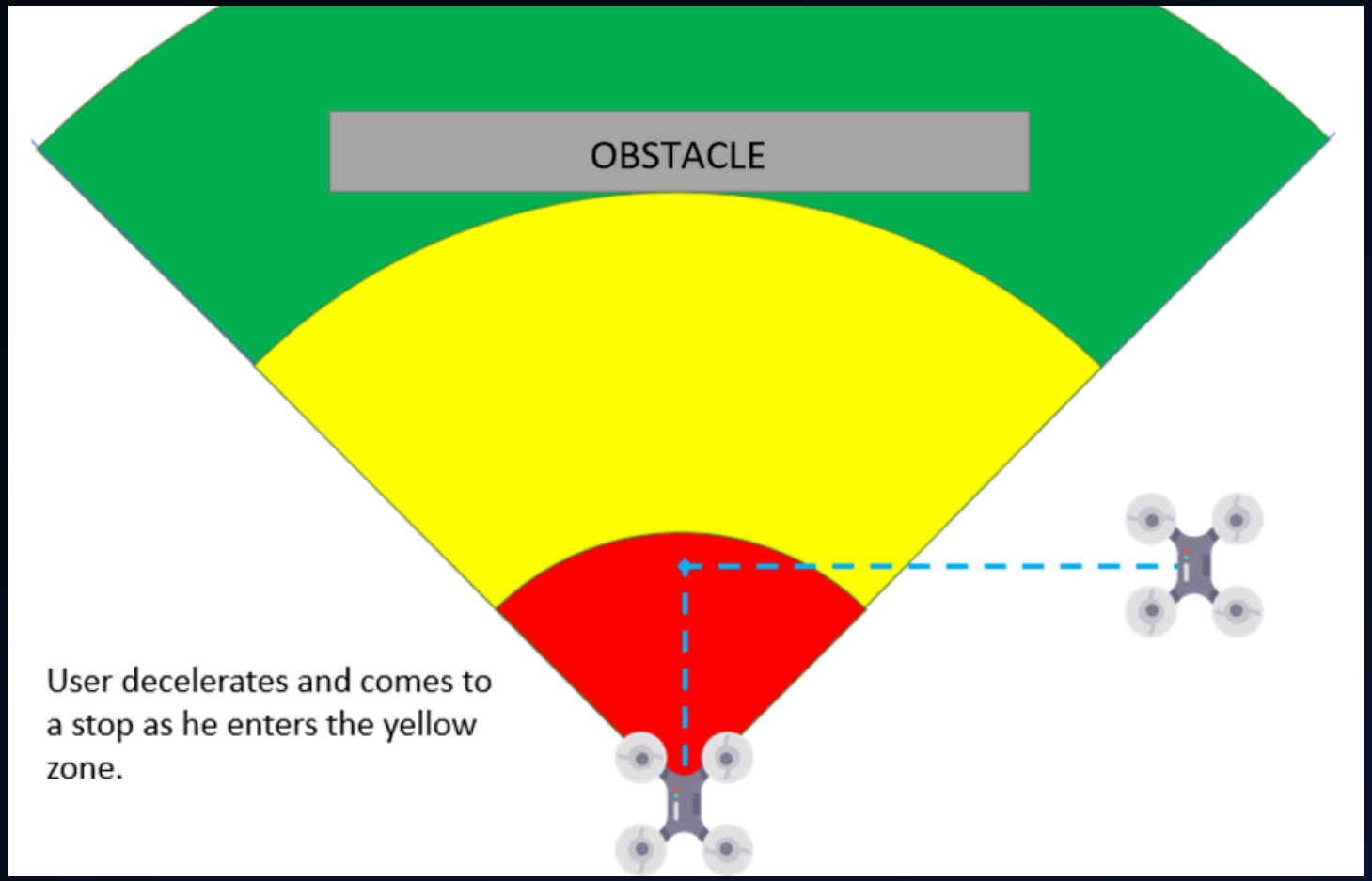
Thank You!

SPECIAL THANKS TO:
Prof. Leonard Kleinrock
Britt Paris

Idea


OBSTACLE
1
2
1. User accelerates inside yellow zone.
2. User maintains constant speed in yellow zone.

Idea


OBSTACLE
User decelerates and comes to a stop as he enters the yellow zone.

Idea