FlySense

Augmented Reality FPV assisted navigation (applied to a helicopter)

Design Review
29th January 2018

Shivang Baveja
Nick Crispie
Joao Fonseca Reis
Harikrishnan Suresh
Sai Nihar Tadichetty

The final SVE system will have three major components: Aerial, Communications & User System

Aerial System

- DJI Matrice 100 (mounted with Jetson TX-2, Velodyne VLP16 Puck, PDB) and FPV Camera

Communications

- 5 GHz Dual Radio Base Station with MIMO technology

User System

- Epson BT 300 Augmented Reality headset
- Headset for audio warnings & voice command recognition

Mandatory Functional and Performance Requirements

Feature	The system SHALL	Target Performance
Input	• Receive sensor state variable data (pose estimate, LIDAR input)	• Receive Point cloud from 1 Velodyne VLP-16 • Receive pose estimates from DJI M100
	• Receive Voice commands to toggle through FlySense widgets (Heads-up-display, Bird's eye view)	• 5 commands • 90% recognition without noise • 70% accuracy with noise
Process / Plan	• Detect obstacles in flight envelope	• Projected 5 seconds into future • 2m X 2m in distances less than 10m
	• Generate bird's eye view of obstacles surrounding the vehicle	• Image generated in vehicle frame >=10Hz
	• Color obstacles in bird's eye view	• Into Red, Yellow or Green based on time to impact, pilot's inputs
	• Override pilot commands to prevent collision	• Stop the aerial system 1m before the obstacle
Output / Convey	• Render HUD, horizon	• >10 Hz refresh rate
	• Render Bird's Eye View	• >10 Hz refresh rate
	• Generate Sound warnings	• Obstacle in flight path with least time to impact • Binary audio, Left or Right based on obstacle • Latency less than 1sec

Desired Functional and Performance Requirements

Feature	The system SHALL	Target Performance
Input	• Voice recognition personalized to User	• Voice command personalized to 3 user
Process / Plan	• Override the pilot to avoid obstacles	• Avoid obstacles by with radial clearance of 2m
Output / Convey	• FPV video overlay on Epson • Segment obstacles	• >10Hz frame rate • Into 2 categories (Trees or building)
	• Recommend feasible trajectory around obstacle	• Avoid obstacle(s) by 1m • Reduce errors by 20% w.r.t. pilot flying w/o FlySense

Updated Non-Functional and Performance Requirements

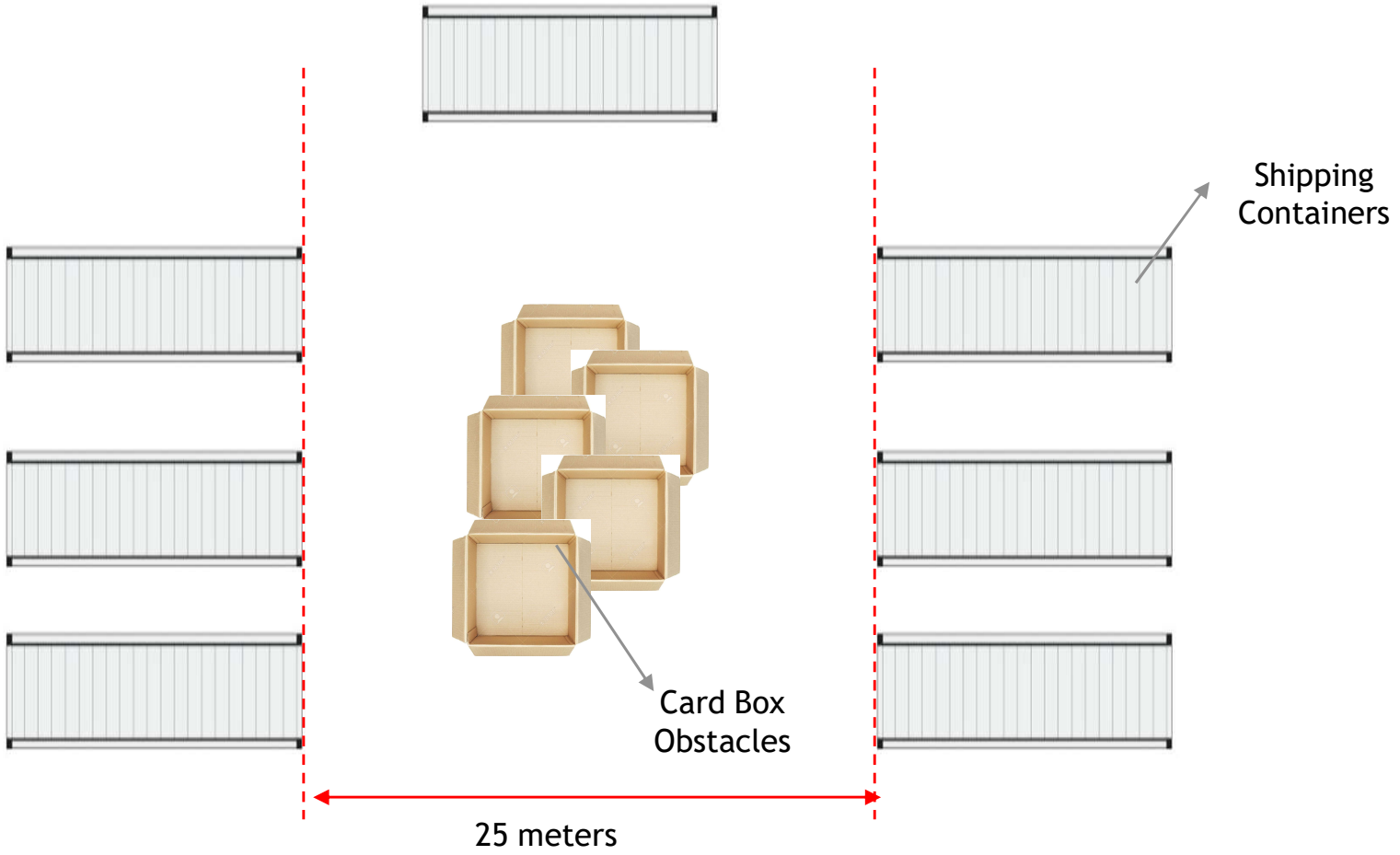
Segmentation	The system WILL	Target Performance
Installation	• Be easy to setup (hardware and software)	• The system will be set up within 1 minute with a single operator
Interaction with Pilot	• Feel natural to the pilot • Be easy to put/remove headwear • Be comfortable to wear headwear for long periods of time	• Focal distance up to 20 meters • Wearable like normal glasses • Weights less than 1 pound
Information Displayed	• Be clear and simple • Be non intrusive to the pilot • Be non distracting for the pilot	• Focus group with 3 pilots using solution
Other criteria	• Be substantially more affordable than available solutions (e.g. fighter jet pilot helmets)	• Solution hardware cost below USD 5,000

Tentative

First we will attach a long tail to the quadcopter to simulate the behavior of a helicopter!

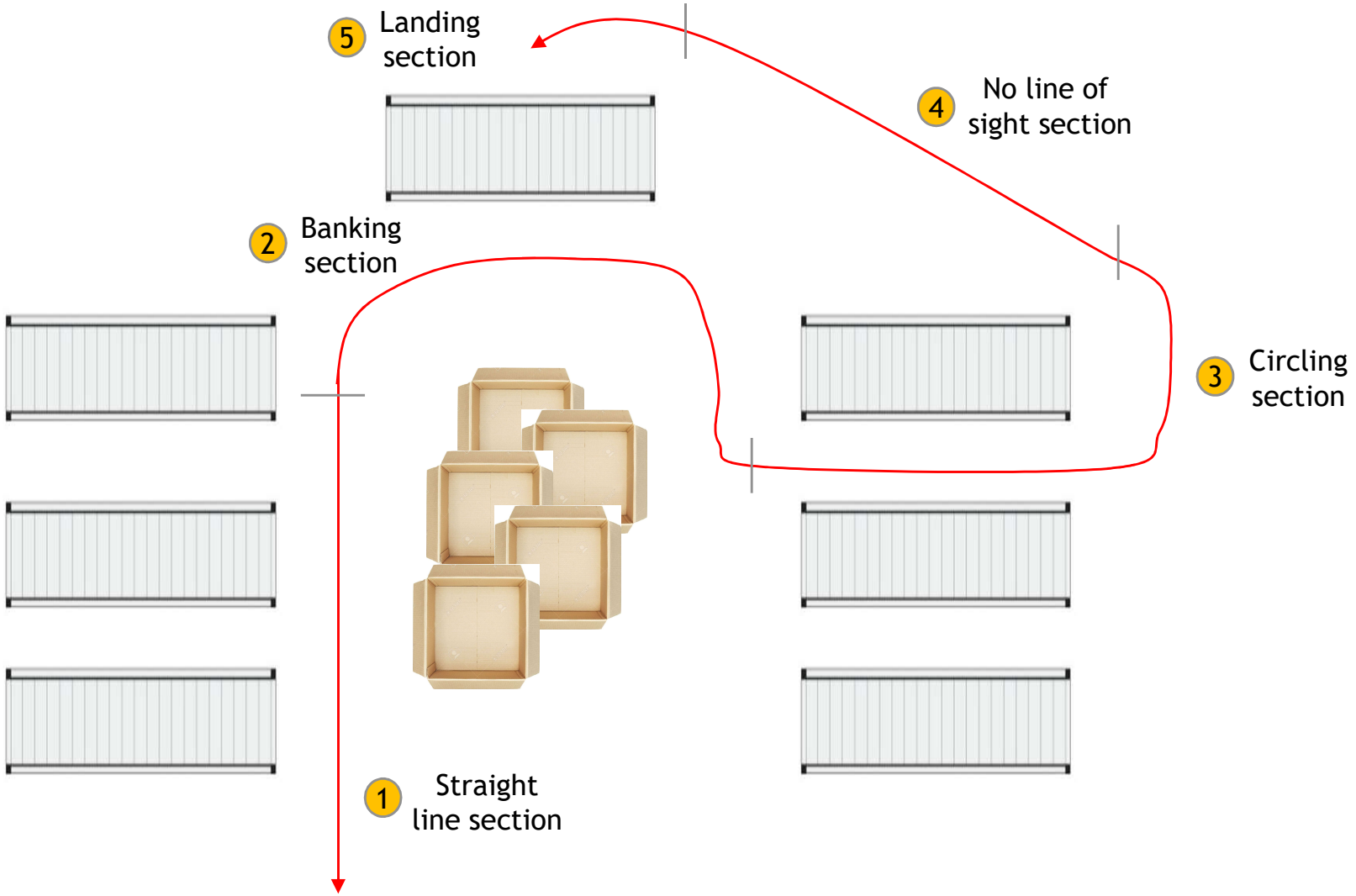
Tentative

The maze test at NEA testing field will be done both with and without the FlySense system



Tentative

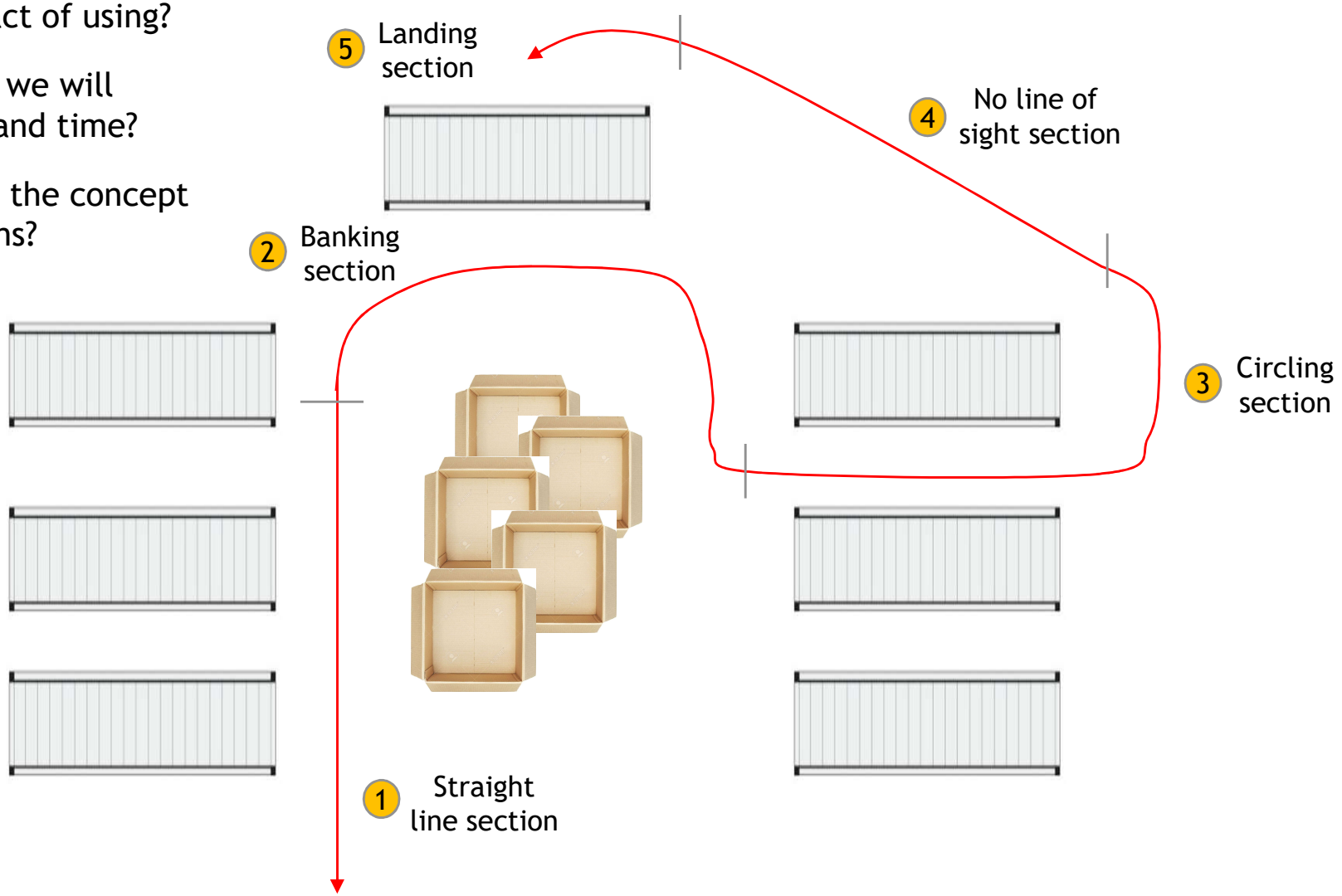
The maze test at NEA testing field will be done both with and without the FlySense system



Tentative

The maze test at NEA testing field will be done both with and without the FlySense system

- Do you agree that we measure the marginal impact of using?
- Do you agree that we will measure # errors and time?
- Do you agree with the concept of multiple sections?



Tentative

The “virtual obstacle” test will be done in open field

- How best to mark the virtual wall in the real world?
- How best to monitor whether or not the red line is surpassed?

1 Quad flying at 5 meters altitude

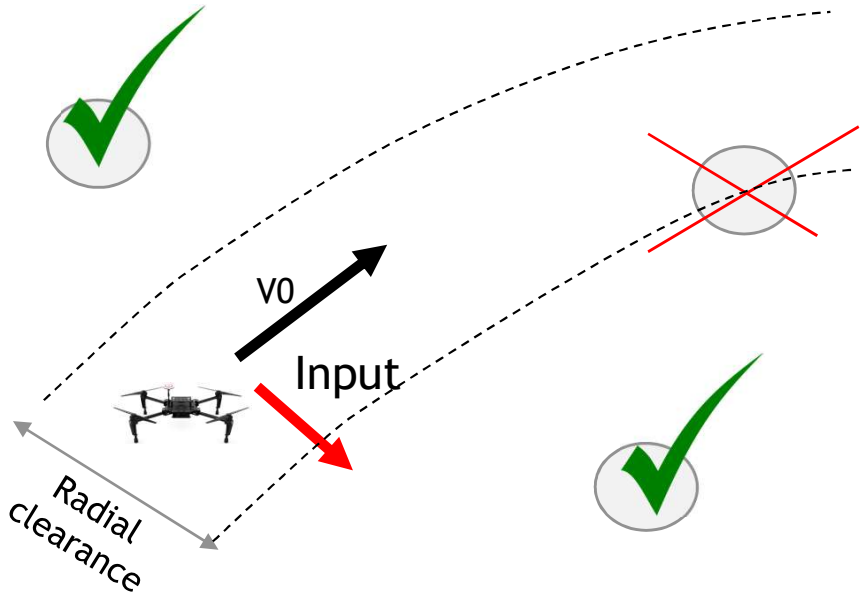
2 Pilot tries to crash into the virtual wall added in the LIDAR data

Virtual Wall

For discussion

The objective of this feature is simply to prevent obvious disasters by stopping

What we plan to do



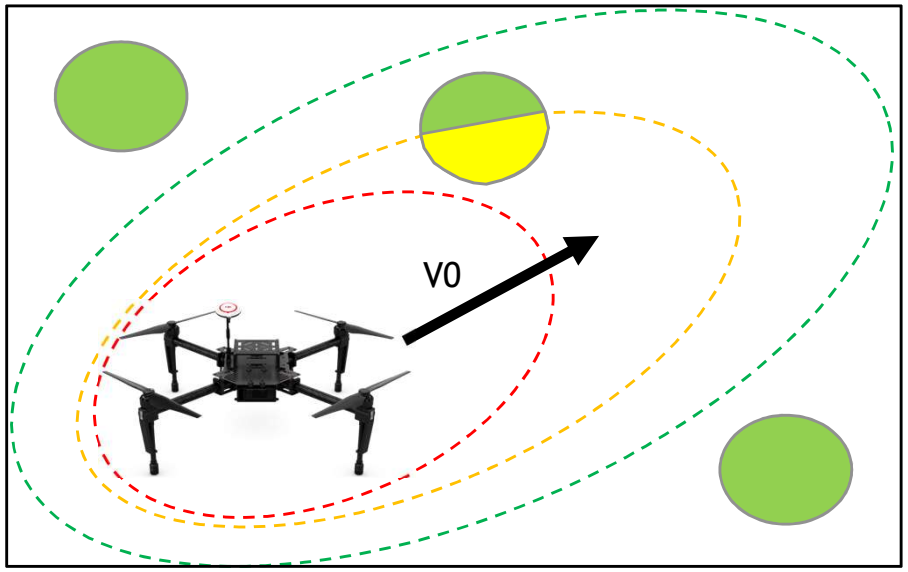
Questions for you

- Does anyone have experience in overriding pilot inputs in DJI quadcopters?
- We have found a logarithmic controller that does this task efficiently (non PID), but are not sure how to treat the pilot input (smooth it?)

For discussion

The objective of the flight envelope coloring is to alert the pilot of what he/she can do…

What we plan to do



- Area of interest selected based on dynamic window computed with vehicle dynamics
- Coloring based on maximum possible pilot input and vehicle dynamics
- Coloring based on δ time for obstacle to be inside the flight envelope [(x/a)^2/+(y/b)^2+(z/c)^2<1]

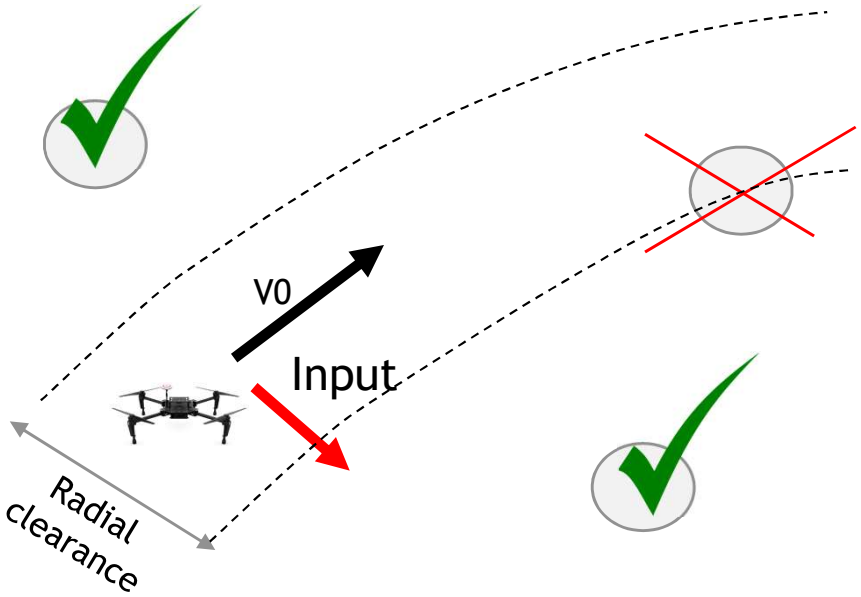
Questions for you

- Is the concept “maximum” input the correct one (or should we restrain it to what the pilot is actually doing right now?)
- Will it be intuitive for the pilot to have a single obstacle with multiple colors?

For discussion

… while the objective of the sound warnings is to alert the pilot of what he/she does not want to do

What we plan to do



Time to impact	Time between beeps
4 to 5.5 seconds	0.5 seconds
2 to 4 seconds	0.3 seconds
0 to 2 seconds	0.1 seconds

Questions for you

- Do you agree with the concept that sound effects need to be less intrusive and thus depend on what the pilot is doing?
- Do we determine left/right based on current speed direction or based on a preferred direction (e.g. x)?

The objective is to improve recognition efficiency to 90% in noisy environments

What we plan to do:

1. Implement a deep neural network/ deep belief network to classify speech into 5 categories: Computer, Alpha, Bravo, Charlie, Close.

What questions we have?

1. Is it a good idea to build your own dataset of words?
2. Has anyone built a speech recognition system before?
3. Any challenges that we should be aware of?

Thank You!