Advancing air mobility via optimization and AI

Yanchao Liu

Department of Industrial & Systems Engineering

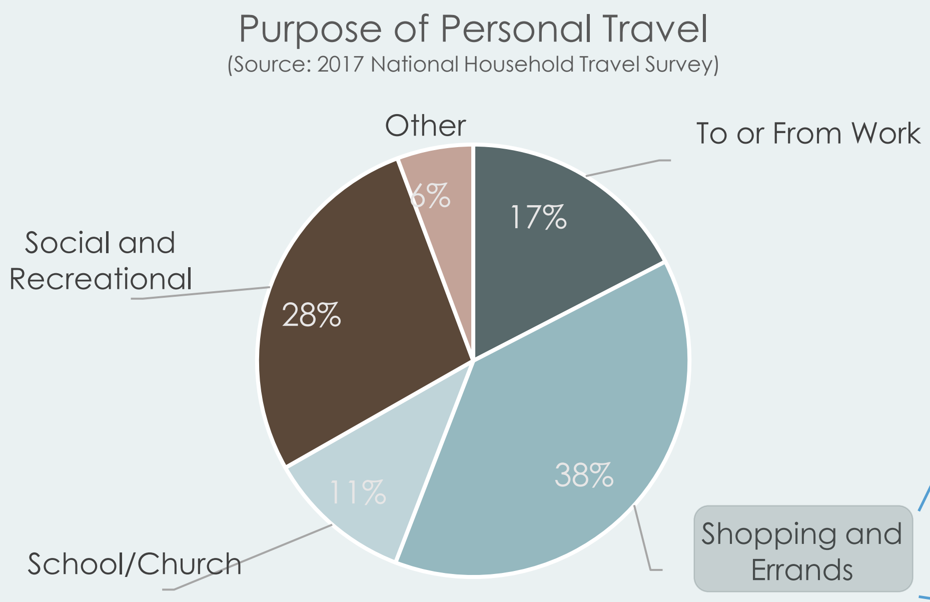
Wayne State University

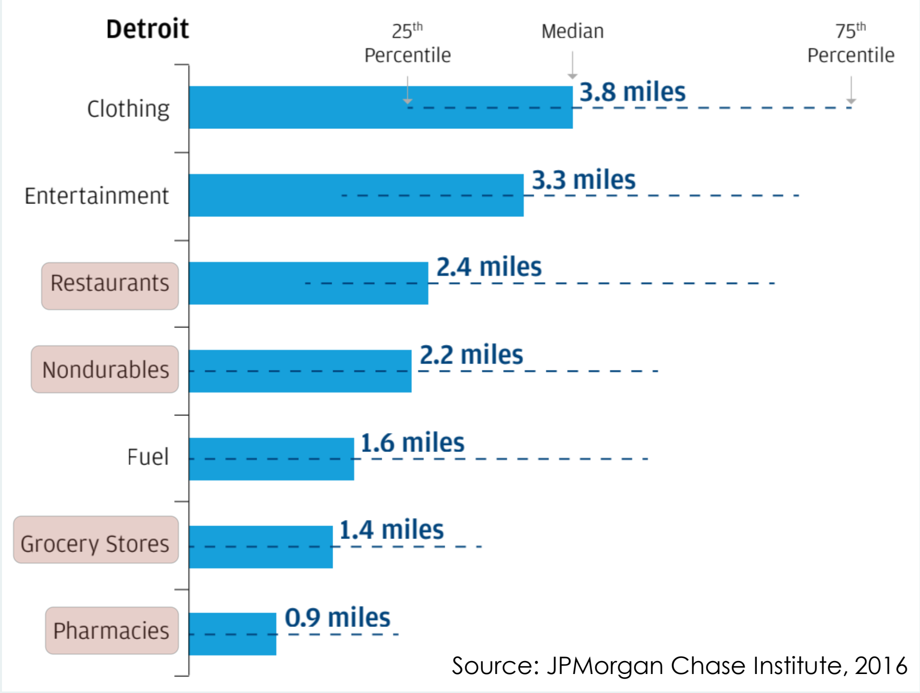
Computation, AI and Data Science (CAD) seminar
9/11/2024

Agenda

- Motivation
- Depot location
- Path planning
- Remote diagnosis
- Autonomous landing
- Future work

What do we travel for?





<u>**Problem**</u>: Most % of personal trips are for "**fetching stuff**", cost time, energy & health.

- Short, boring, dreadful trips
- Creating excessive congestion, pollution, accidents, stress and wastes of time

Autonomy is the way to go in package delivery

- **21.2 billion** packages were shipped in the U.S. in 2022
- From 2017 to 2022, the number of packages the average American received in a year **increased by 73%**

"Finding enough labor for the logistics industry could become extremely difficult or even impossible."
-- DHL

"Get ready for a world where autonomous vehicles deliver 80 percent of parcels." -- McKinsey&Co

[Package Delivery Statistics (2024): per Day, Month & Year (capitaloneshopping.com)]

Drone delivery is booming

Amazon Uber Wing Walmart

DHL Zipline UPS Antworks

China's Low-Altitude Economy initiative

China's Futuristic Industries: Investment Prospects in the Emerging Low-Altitude Economy

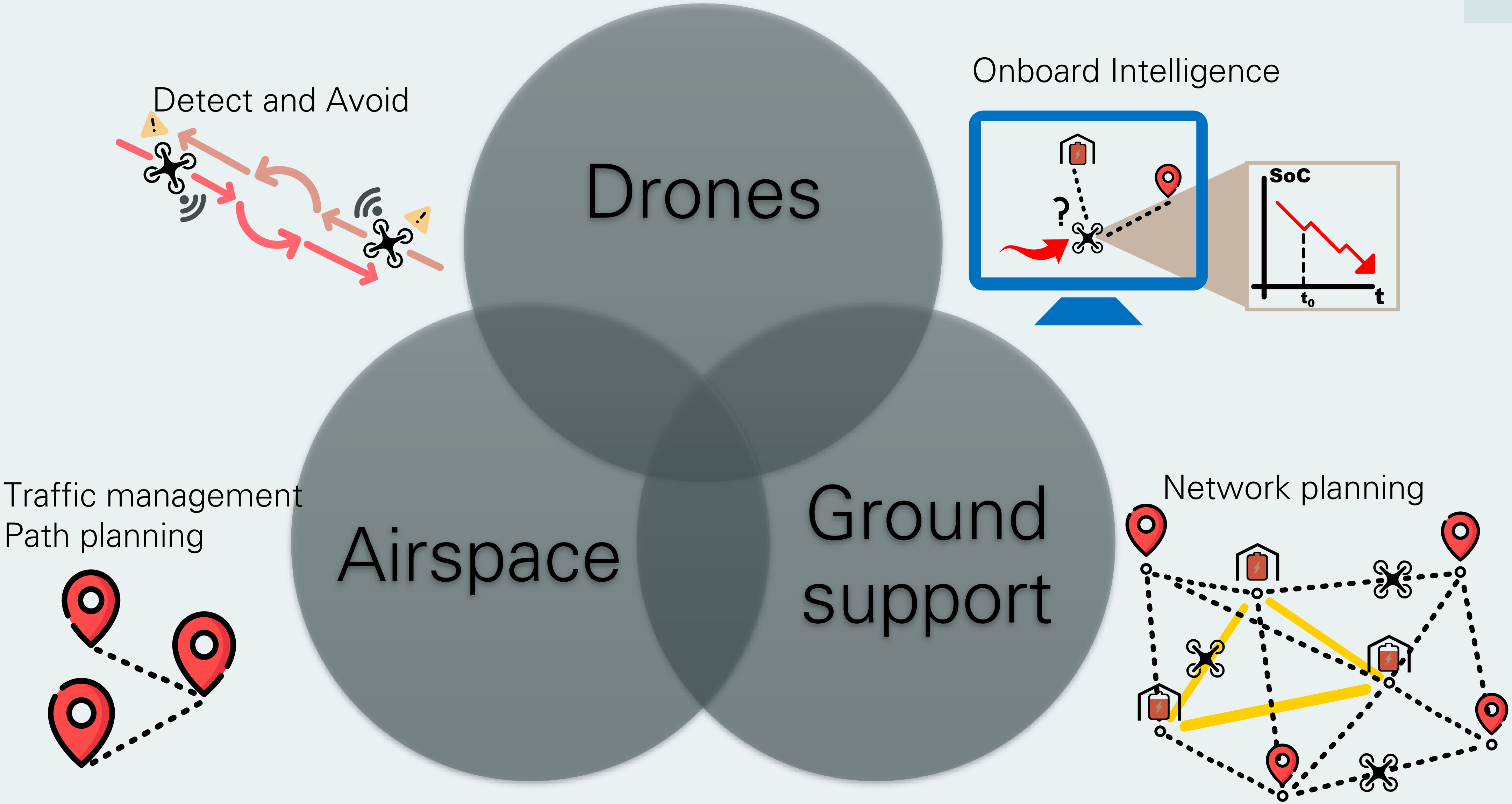
July 24, 2024 Posted by China Briefing Written by Yi Wu and Giulia Interesse Reading Time: 11 minutes

China's low-altitude economy is rapidly growing, driven by supportive policies and technological advancements, with projections indicating a significant economic contribution by 2025. Key regions like Guangdong, Shenzhen, and Chengdu are spearheading development through substantial investments and regulatory support, despite challenges in infrastructure and safety. We examine the investment opportunities and business outlook of this burgeoning sector

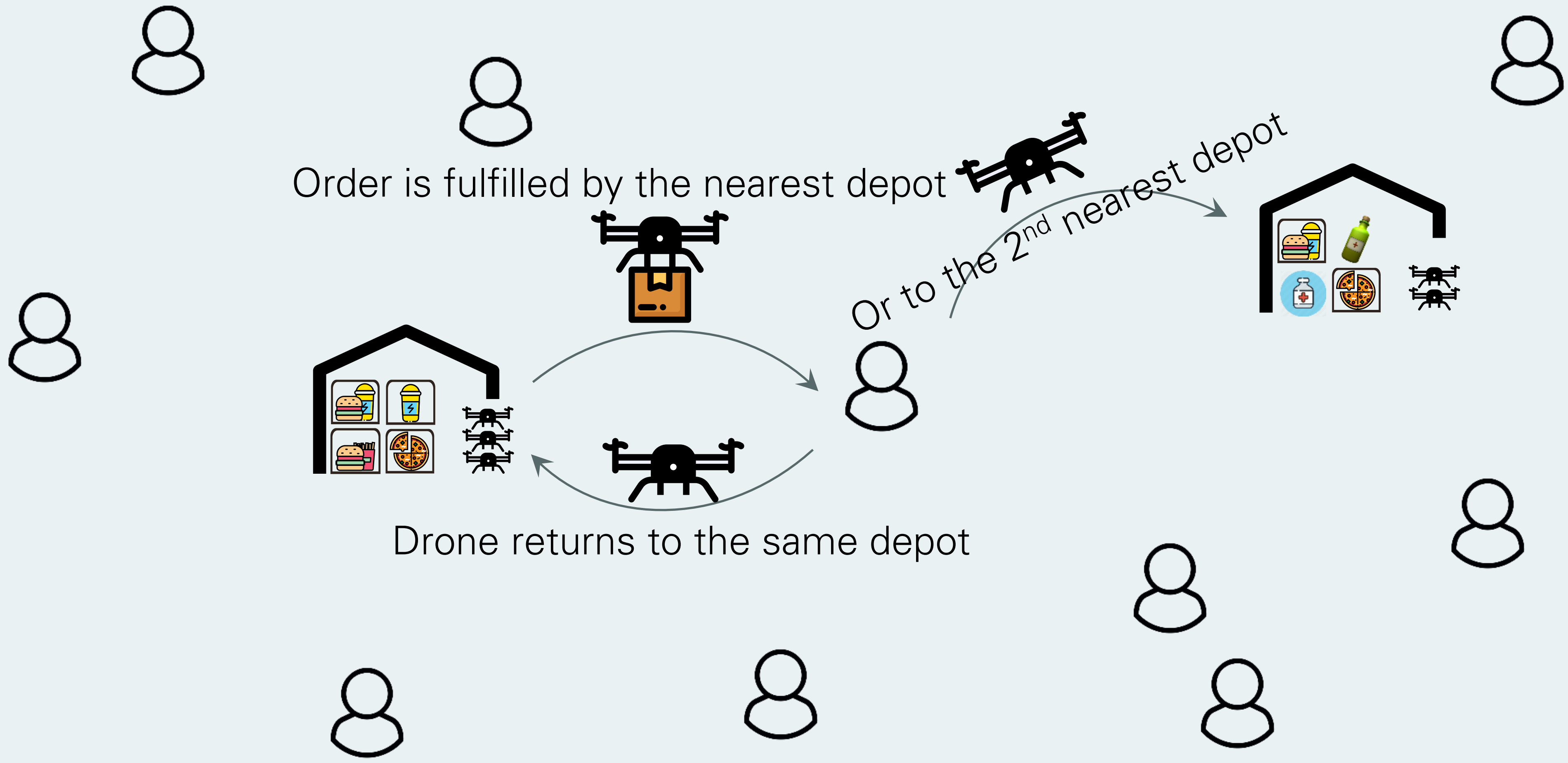
During the annual Central Economic Work Conference, which concluded on December 18, 2023, Chinese policymakers outlined priorities for 2024 economic work. The meeting identified the low-altitude economy as a strategic emerging sector, alongside key industries such as bio-manufacturing and commercial aerospace innovation.

The term "low-altitude economy" refers to a spectrum of economic activities occurring within low altitude airspace, defined as the space 1,000 meters above ground. This includes various activities and industries centered around civil-manned and unmanned aerial vehicles, such as passenger transport, cargo delivery, manufacturing, low-altitude flight operations, and integrated services.



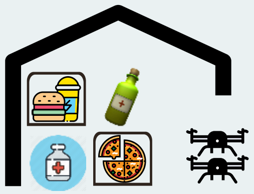

Detect and Avoid
Onboard Intelligence
SoC
t0
t
Drones
Airspace
Ground support
Traffic management
Path planning
Network planning

Where to place the vertiports / depots?

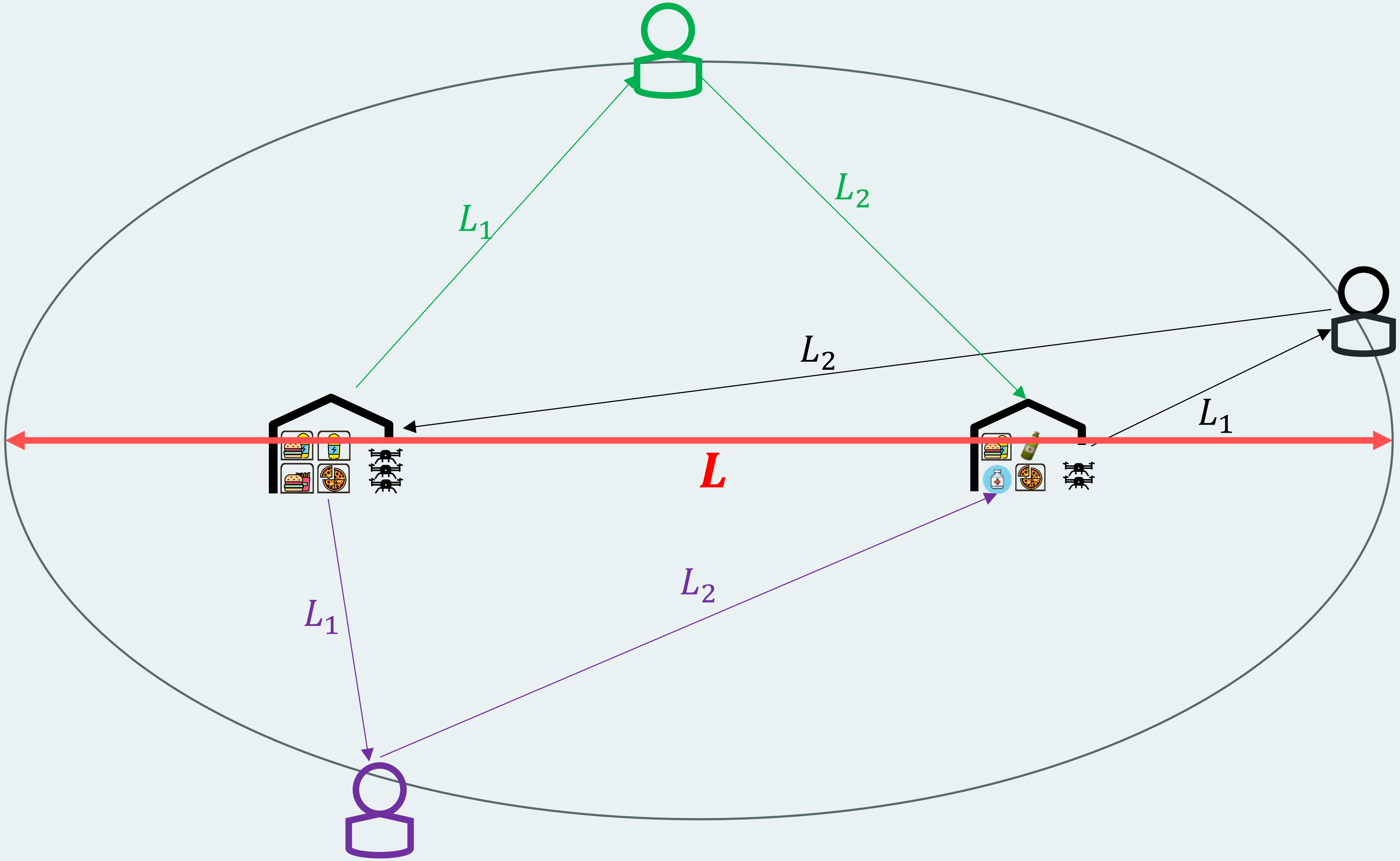


There are n demand centers in a city

Decide the location of p depots so that the longest round-trip distance is minimized.



Minimize the upper bound of battery consumption of a trip.



Given a trip length budget L, with $L_1 + L_2$, the sum of the delivery and return trips, $\leq L$

Customers that are reachable by a pair of depots must lie in the **elliptical area** determined by the depots as focus points and L as the major axis length.

The problem becomes determining the location of p depots so that

- Each customer is covered by an ellipsis
- Each ellipsis's foci are depots
- The *size** of the largest ellipsis is minimized.

* *Size* means major axis length.

Problem Formulation

Given n demand points with coordinates (a_i, b_i), $i = 1, \dots, n$, on a plane, find p depot locations X_j, $j = 1, \dots, p$, in order to minimize

$$\max_{1 \le i \le n} \left\{ \min_{1 \le j \le j' \le p} \left\{ D_i(X_j) + D_i(X_{j'}) \right\} \right\}$$

where $X_j \coloneqq (x_j, y_j)$ is the location of depot j, for $j = 1, \dots, p$, and $D_i(X_j) \coloneqq \sqrt{(x_j - a_i)^2 + (y_j - b_i)^2}$ is the Euclidean distance between demand point i and depot j.

We call this problem ***Euclidean p-Elliptical Cover*** problem.

<u>Mixed Integer Nonlinear Programming (MINLP) formulation (k = 2):</u>

Minimize	L	
Subject to	$L \ge \sum_{j=1}^{p} z_{ij} \left[(x_j - a_i)^2 + (y_j - b_i)^2 \right]^{\frac{1}{2}}$,	for $i = 1, \dots, n$
	$\sum_{j=1}^{p} z_{ij} = k$,	for $i = 1, \dots, n$
	$z_{ij} \in \{0, 1\}$,	for $i = 1, \dots, n$, $j = 1, \dots, p$
	$x_j, y_j \in R$	for $j = 1, \dots, p$

General MINLP formulation:

Minimize	L	
Subject to	$L \geq \sum_{j=1}^{p} z_{ij} \left[\left(x_j - a_i\right)^2 + \left(y_j - b_i\right)^2 \right]^{\frac{1}{2}},$	for $i = 1, \dots, n$
	$\sum_{j=1}^{p} z_{ij} = k,$	for $i = 1, \dots, n$
	$z_{ij} \in \{0, 1\},$	for $i = 1, \dots, n,\ j = 1, \dots, p$
	$x_j, y_j \in R$	for $j = 1, \dots, p$

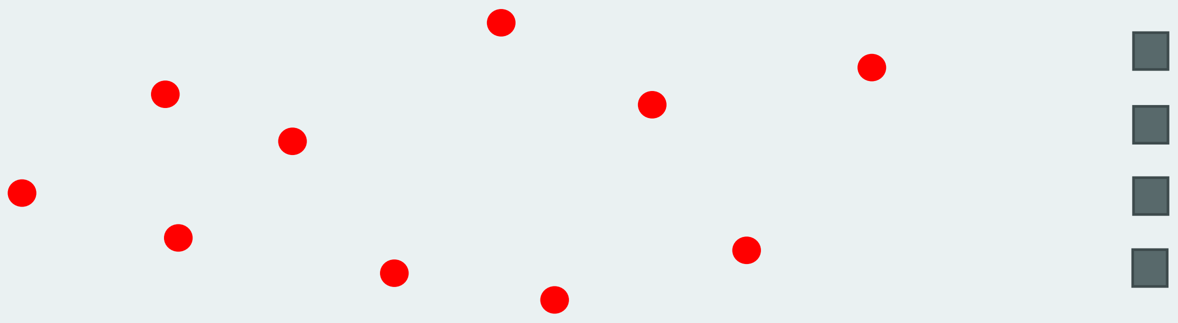
When $k = 1$, it is the Euclidean p-center problem. Proven NP-hard. [Megiddo and Supowit, 1984]

When $k = p$, the problem is convex and solvable in polynomial time. See [Blanco and Puerto, 2021].

When $k = 2$, it is the Euclidean p-Elliptical Cover problem. Proven NP-hard, heuristic solution [Liu, 2023].

If the demand-depot assignment z_{ij} is fixed, then the problem is convex, i.e., a second-order cone problem (SOCP), solvable in polynomial time.

Assignment is the difficult part



- $n = 10$ demand points
- $p = 4$ depots
- There are $\binom{4}{2}^{10}$ = 60,466,176 different demand-depot assignments
- Since the depots are unlabeled, the number of unique assignments is 162,575
- Difficulties:
  - Algebraic formulation intrinsically labels objects via indexing, hard to deduplicate
  - The number of unique assignments is still too big to check one by one

A locate-allocate algorithm

Will terminate in finite (usually few) iterations.

Step 1: Initialize $L^* = \infty$ and randomly sample p depot locations on the plane, $X_1, \dots, X_p$

Step 2 (Allocate): For each demand point i, compute the distances $D_i(X_j), j = 1, \dots, p$, and let $J(i) \coloneqq \{j_i, j_i'\}$ where $j_i = \arg\min_{\{1,\dots,p\}} D_i(X_j)$ and $j_i' = \arg\min_{\{1,\dots,p\}\backslash\{j_i\}} D_i(X_j)$.

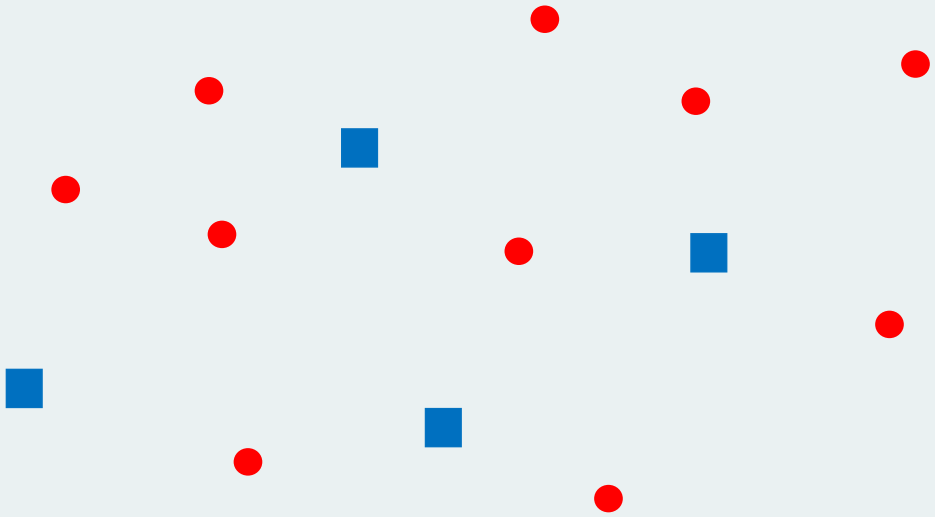
Step 3 (Locate): Solve the SOCP problem:

$$
\begin{array}{lll}
\text{Minimize} & L & \\
\text{Subject to} & L \geq \sum_{j \in J(i)} \left[\left(x_j - a_i\right)^2 + \left(y_j - b_i\right)^2 \right]^{\frac{1}{2}}, & \text{for } i = 1, \dots, n \\
 & x_j, y_j \in R & \text{for } j = 1, \dots, p
\end{array}
$$

to obtain the optimal value $\hat{L}$, and the optimal solution $\hat{X}_j$, $j = 1, \dots, p$.

Step 4: If $\hat{L} - L^* \geq -\epsilon$ (insufficient decrease), stop and return $\{\hat{X}_1, \dots, \hat{X}_p\}$ as solution; otherwise, update $L^* \leftarrow \hat{L}$, $\{X_1, \dots, X_p\} \leftarrow \{\hat{X}_1, \dots, \hat{X}_p\}$, go to Step 2.

Demonstration



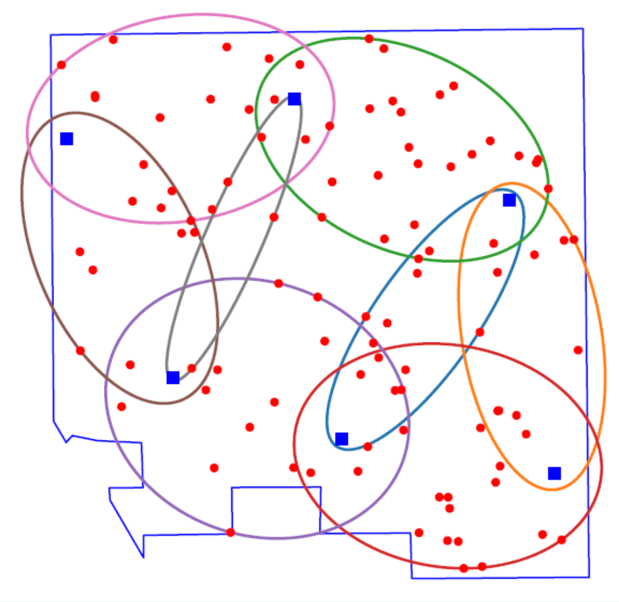
Where to place the four depots?

[Demo (n=10, p=4)]

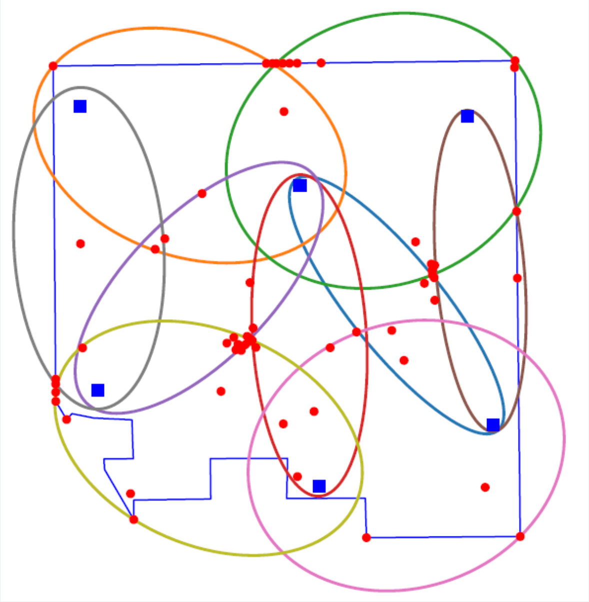
[Demo (n=30, p=4)]

Paper: Y. Liu (2023). An elliptical cover problem in drone delivery network design and its solution algorithms, *European Journal of Operational Research*, vol 304, issue 3, 2023.
Code: GitHub - profyliu/elliptical_cover

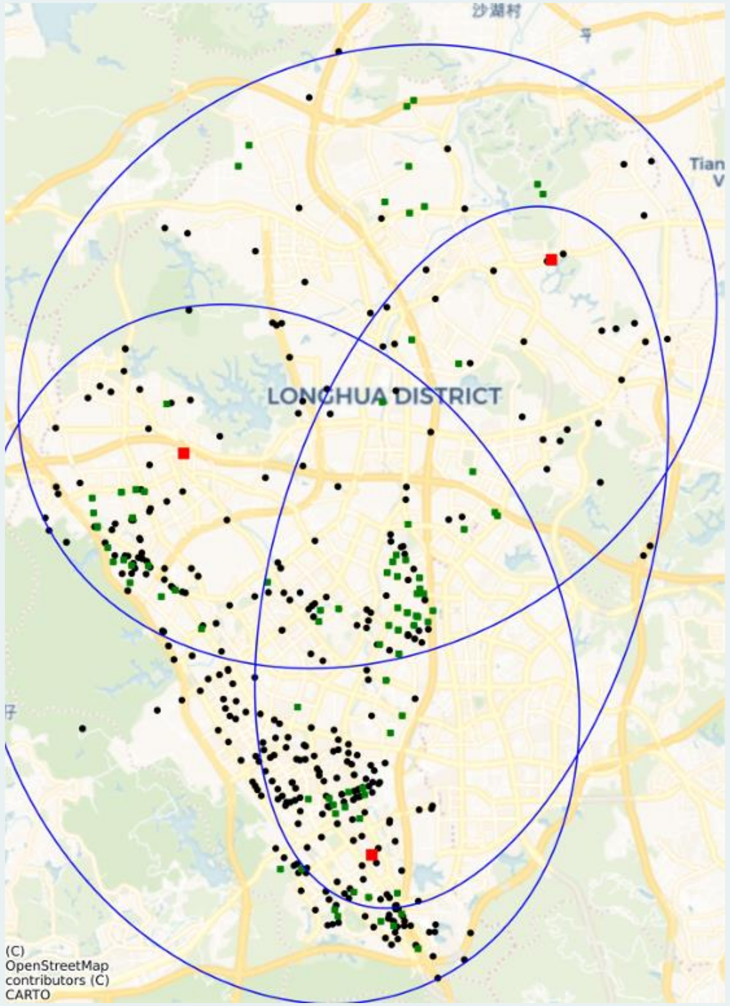
Cover 100 points

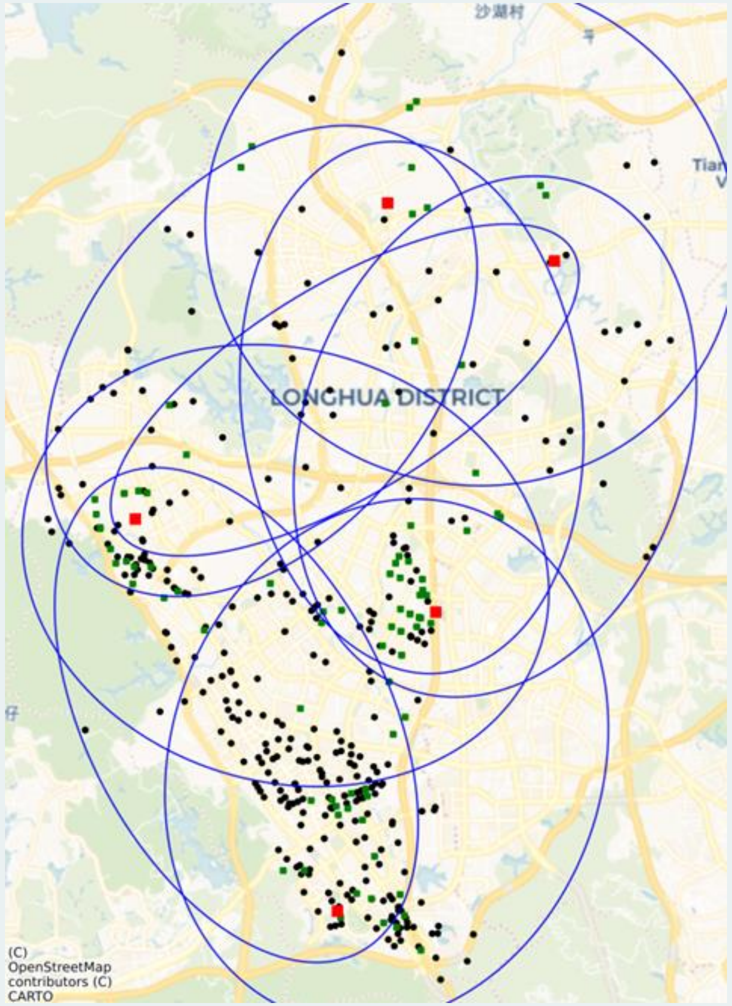


Cover the whole area



Application in planning for Low-altitude Economy in Shenzhen, China

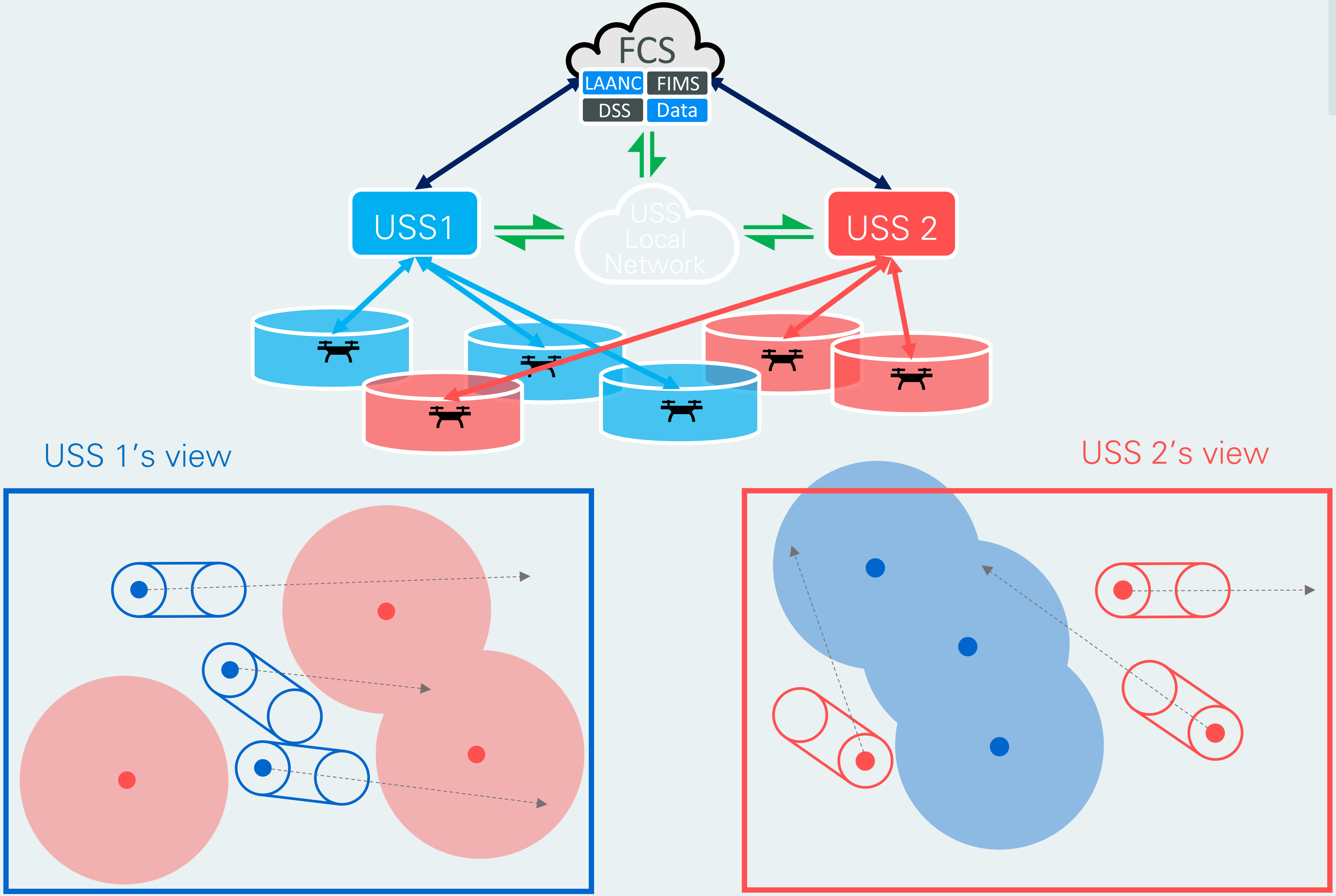




J. Zhao, B. Xie and L. Yu. (2024). An ellipse-based locating method for flexible deployment of emergency UAVs, *Socio-Economic Planning Sciences*.

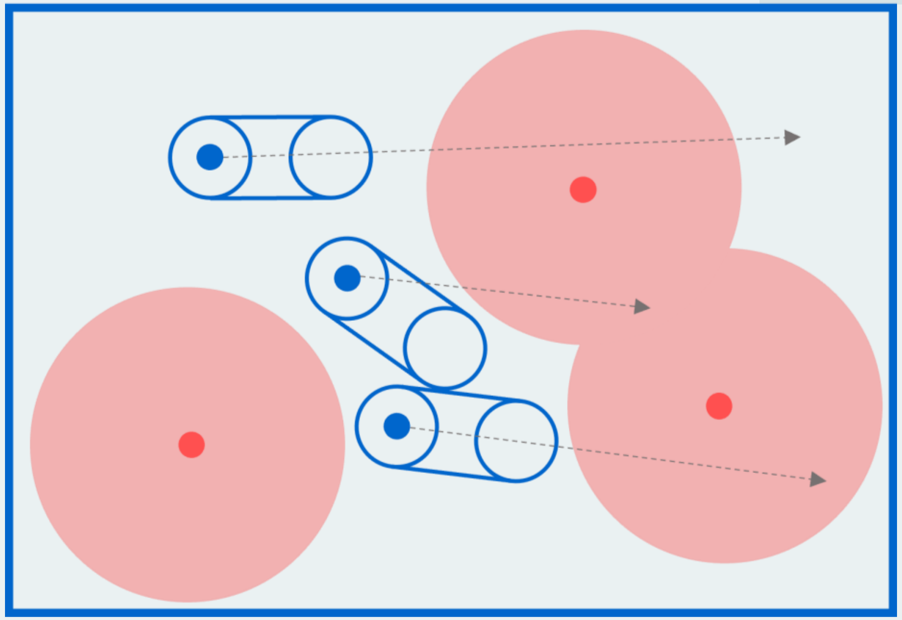
How to route the air fleet?

When there are other, non-cooperative UAVs traversing the same airspace



FCS
LAANC
FIMS
DSS
Data
USS1
USS
Local
Network
USS 2
USS 1's view
USS 2's view

Agent's Semi-Cooperative Path Planning Model



Decision Variables:

$u_{i,t}$	location of vehicle i at time t
v_i	velocity vector of vehicle i
$w_{i,j,t}$	loss of separation between vehicles i and j at time t

Parameters:

V_i	maximum speed of vehicle i
D_i	destination coordinate of vehicle i
$S_{i,j}$	**intra-fleet** separation between $i, j \in O$
$S'_{i,j}$	**inter-fleet** separation between $i \in O$ and $j \in E$
α_i, β	priorities of vehicle i and a penalty factor
$\hat{u}_{i,t}$	expected location of external vehicle i at time t

Minimize $\sum_{i \in O} \alpha_i \cdot \left(\sum_{t \in T} \lVert u_{i,t} - D_i \rVert\right) + \beta \cdot \sum_{i \in O, j \in E} \left(\sum_{t \in T} w_{i,j,t}^2\right)$

Subject to

$$
\begin{aligned}
& u_{i,t} = u_{i,t-1} + V_i v_i, && \forall i \in O, t \in T \\
& \lVert v_i \rVert \leq 1, && \forall i \in O, t \in T \\
& \lVert u_{i,t} - u_{j,t'} \rVert \geq S_{i,j}, && \forall i, j \in O,\ t, t' \in T \\
& \lVert u_{i,t} - \hat{u}_{j,t} \rVert + w_{i,j,t} \geq S'_{i,j}, && \forall i \in O, j \in E, t \in T \\
& w_{i,j,t} \geq 0, && \forall i \in O, j \in E, t \in T
\end{aligned}
$$

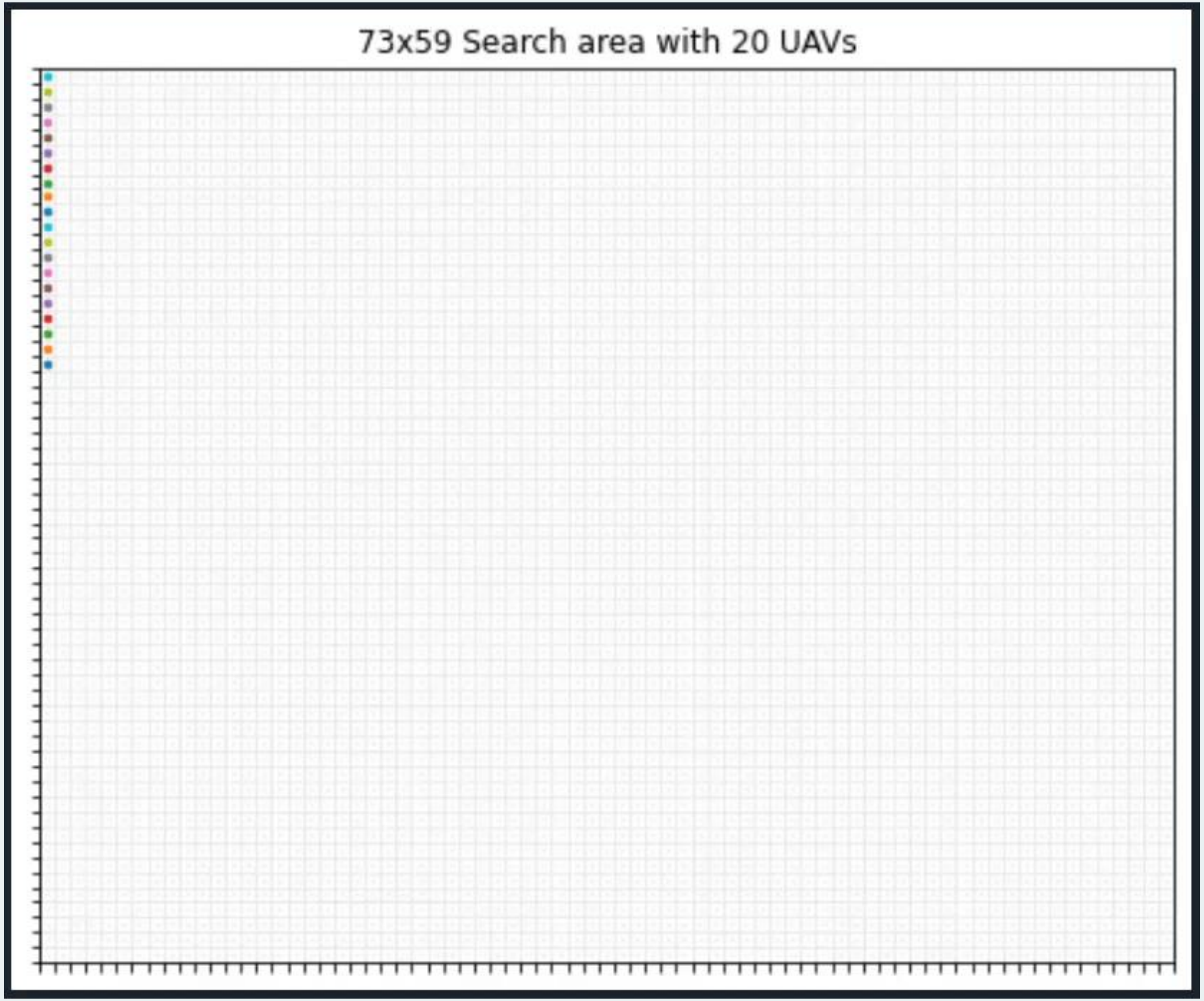
Separation and deadlock resolution can be achieved via setting / adjusting the parameters

Simulation

- [3 vehicles 1 agent]
- [20 vehicles 1 agent]
- [20 vehicles 20 agents]
- [30 vehicles 2 agents]

Y. Liu (2021). A Multi-agent Semi-cooperative Unmanned Air Traffic Management Model with Separation Assurance, *EURO Journal on Transportation and Logistics,* vol 10, 2021.

Y. Liu (2019). A Progressive Motion Planning Algorithm and Traffic Flow Analysis for High-Density 2D Traffic, *Transportation Science*, vol 53, no 6, 2019.



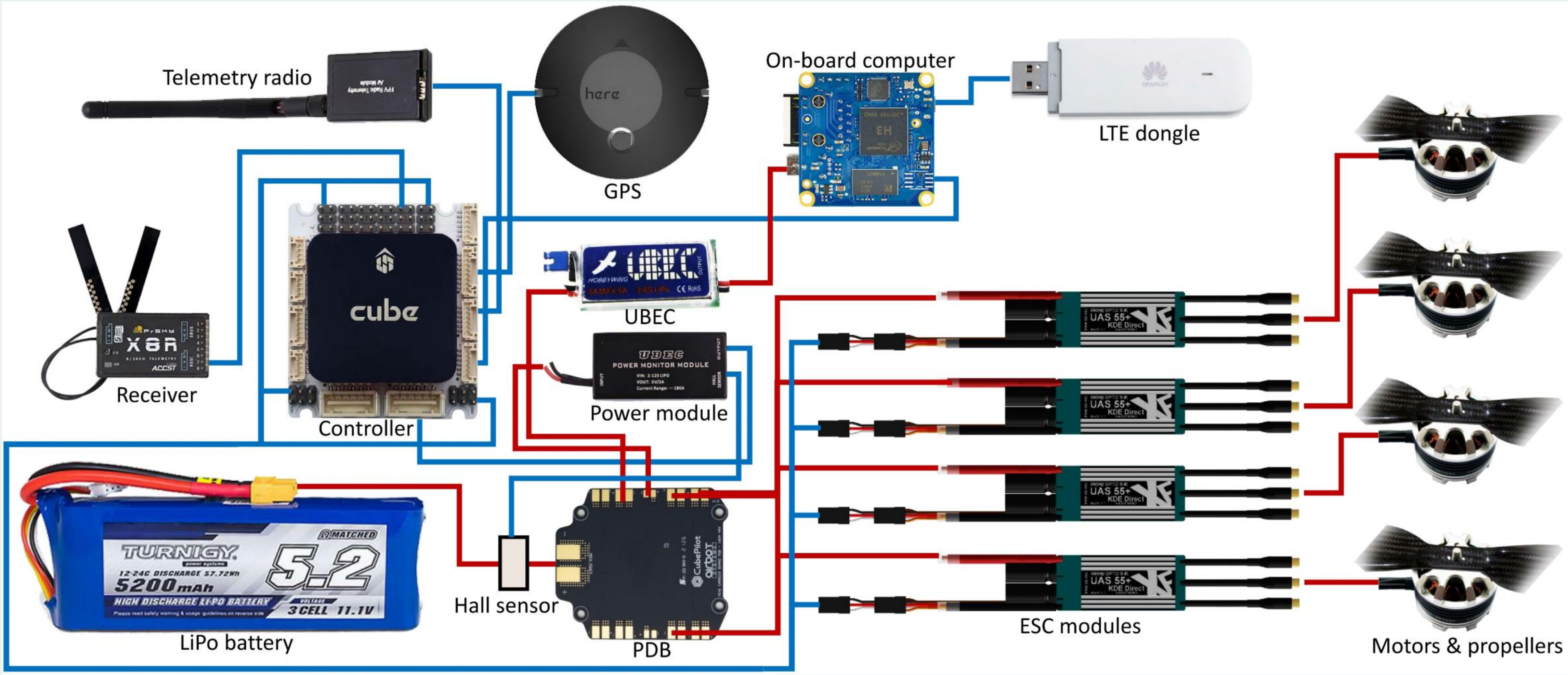
Search an area with n UAVs in minimal time under windy conditions

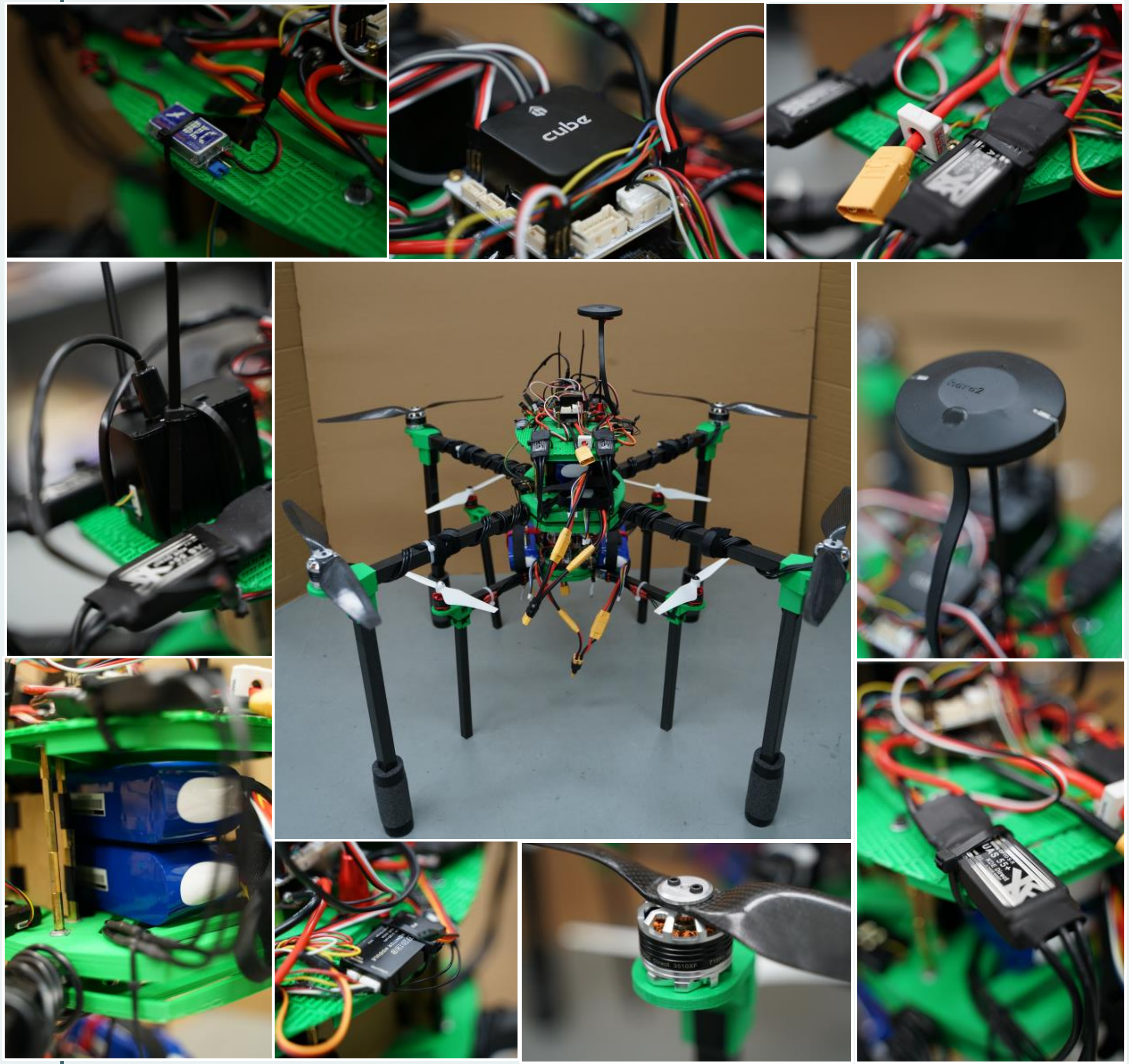
Sina demonstrating the work to visitors from Wuhan Univ. of Technology

S. Kazemdehbashi and Y. Liu (2024). An algorithm with exact bounds for coverage path planning in UAV-based search and rescue under windy conditions, *Computers and Operations Research*, Volume 173.

The Drones

Component wiring of the quadcopter platform used in this research

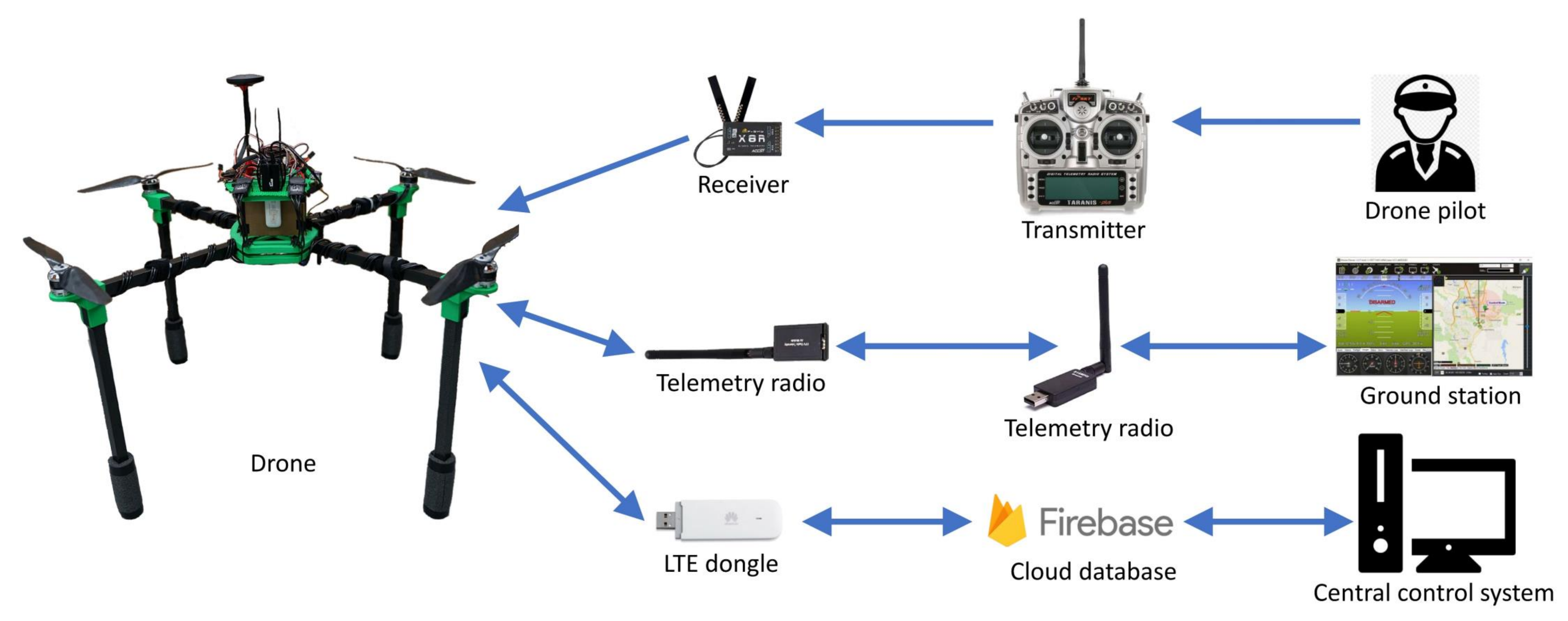




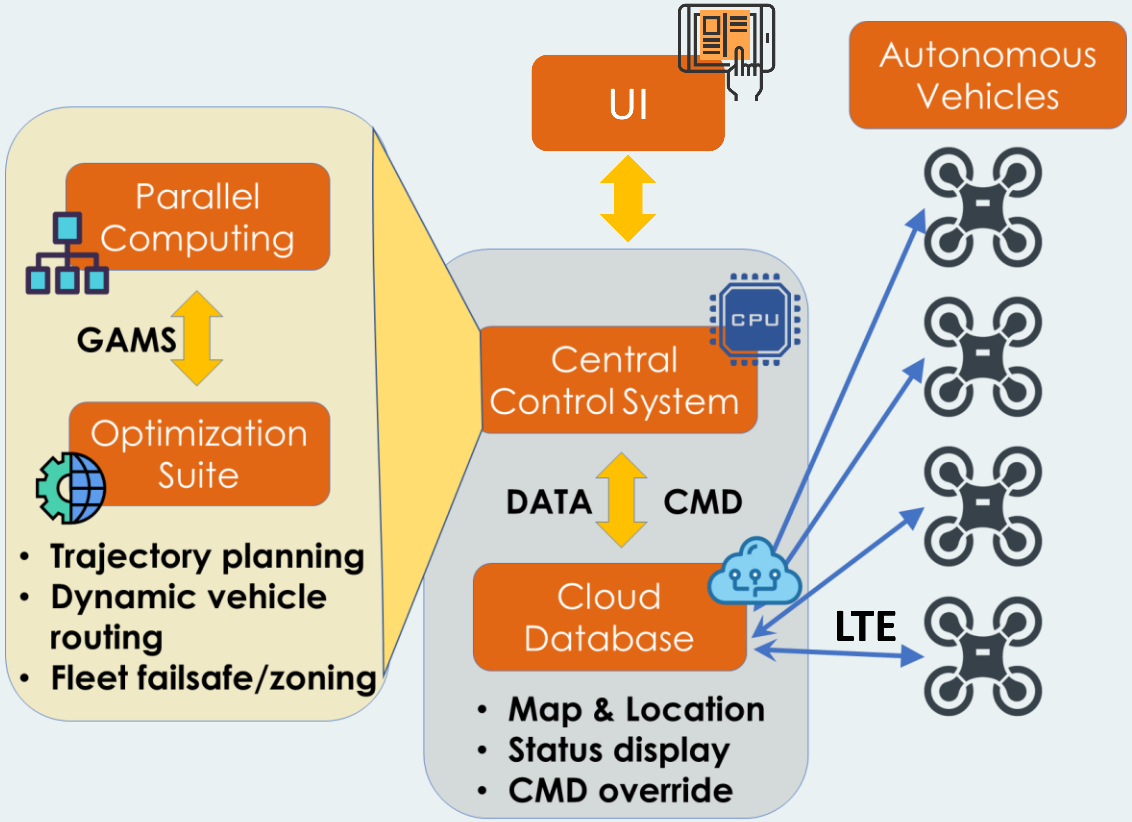
Quadcopter with ArduPilot autopilot system

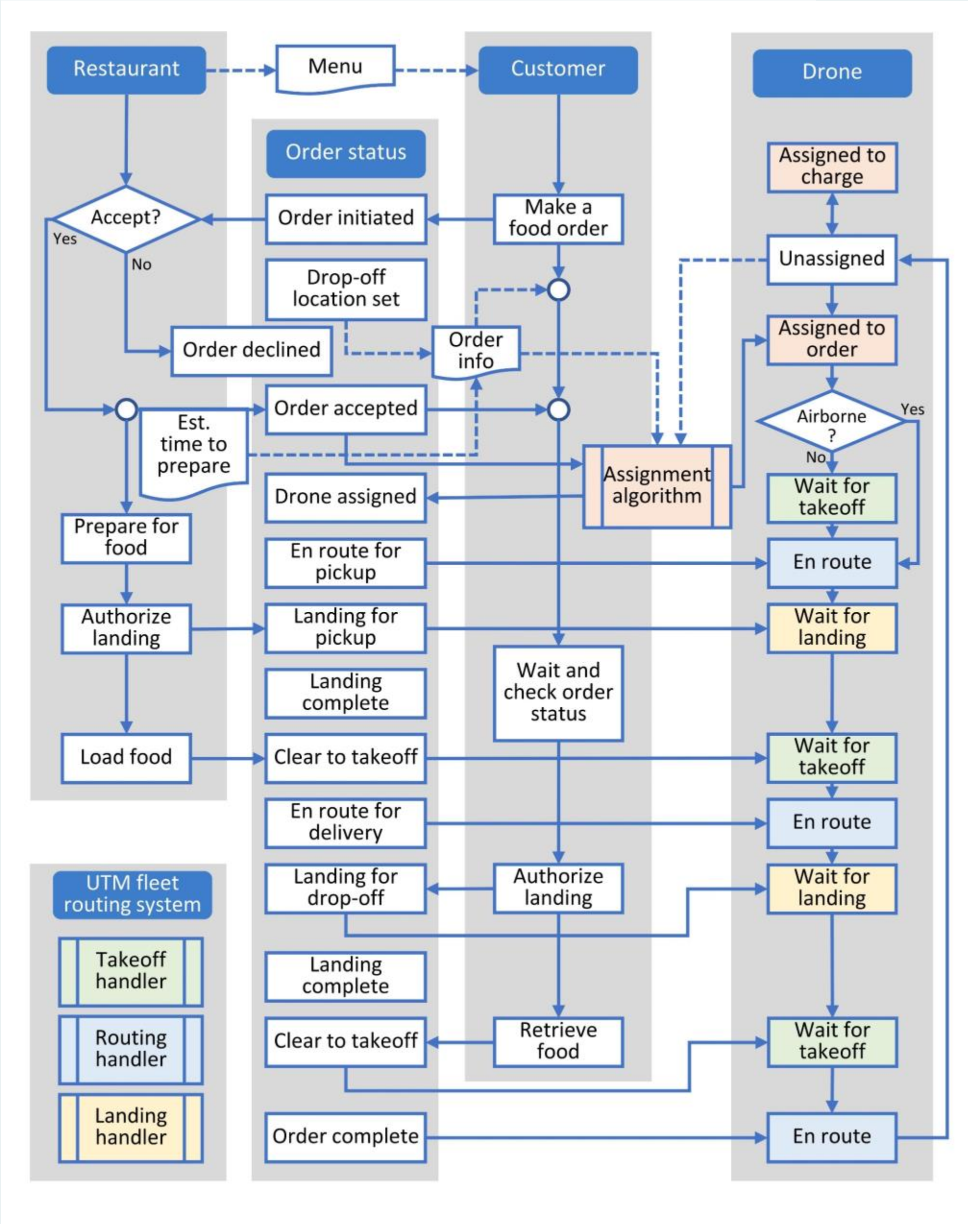
Simulated drones running with ArduPilot SITL (Software in the Loop)

Multiple command and control data links



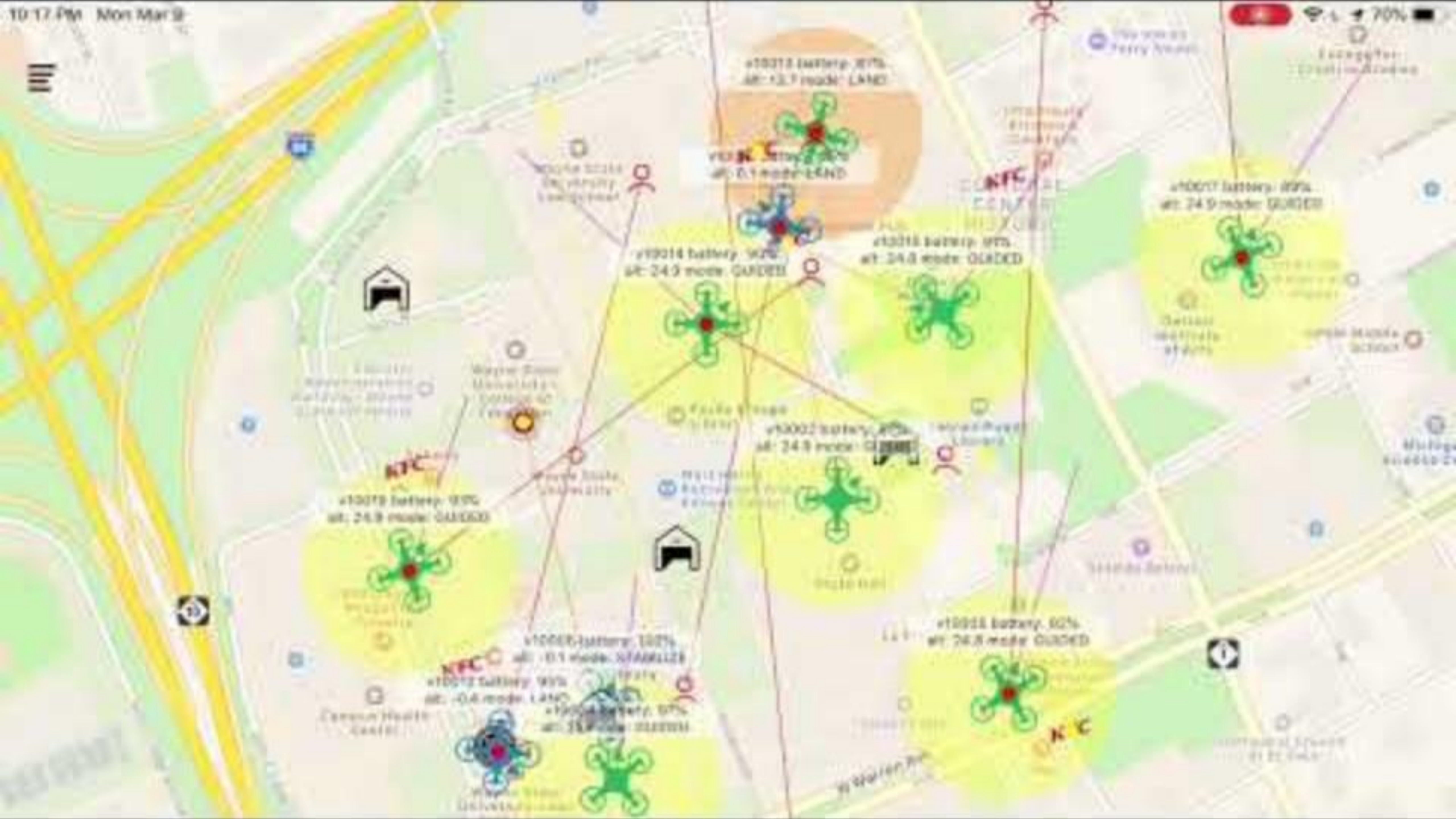
A system architecture for drone delivery services





Y. Liu (2019). An Optimization-driven Dynamic Vehicle Routing Algorithm for On-demand Meal Delivery Using Drones, *Computers and Operations Research*, vol 111, 2019

Z. Zhou and Y. Liu (2022). A scalable cloud-based UAV fleet management system, *Proceedings of FAIM 2022*


10:17 PM Mon Mar 9

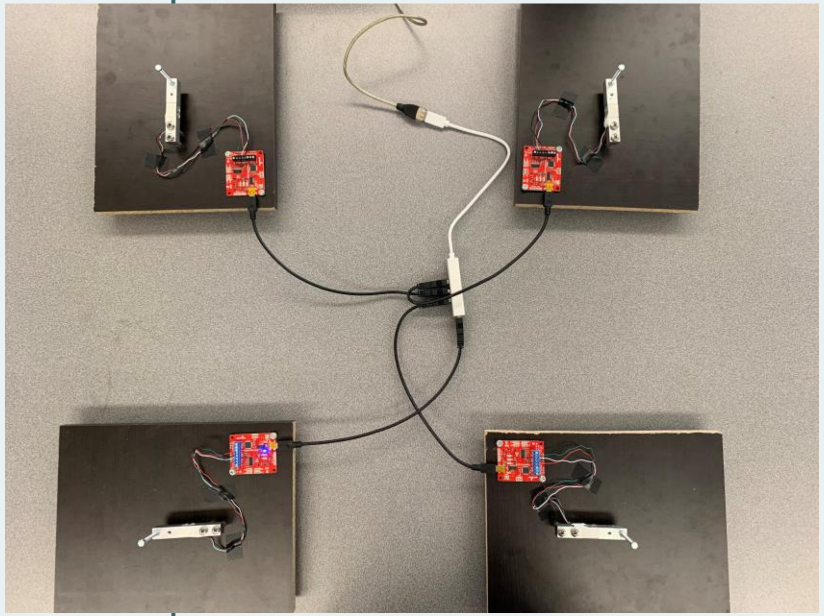
Components can wear and tear after a flight mission

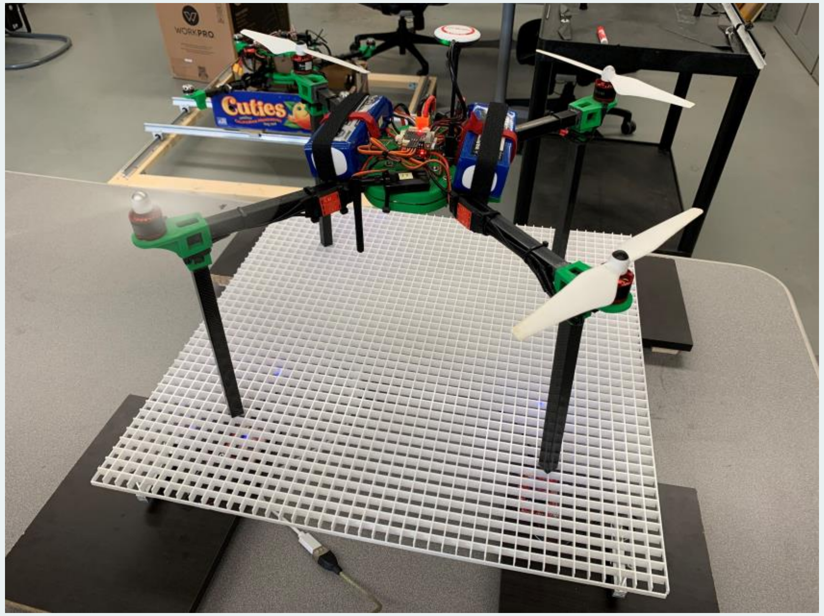
Preflight check is essential to ensure safety

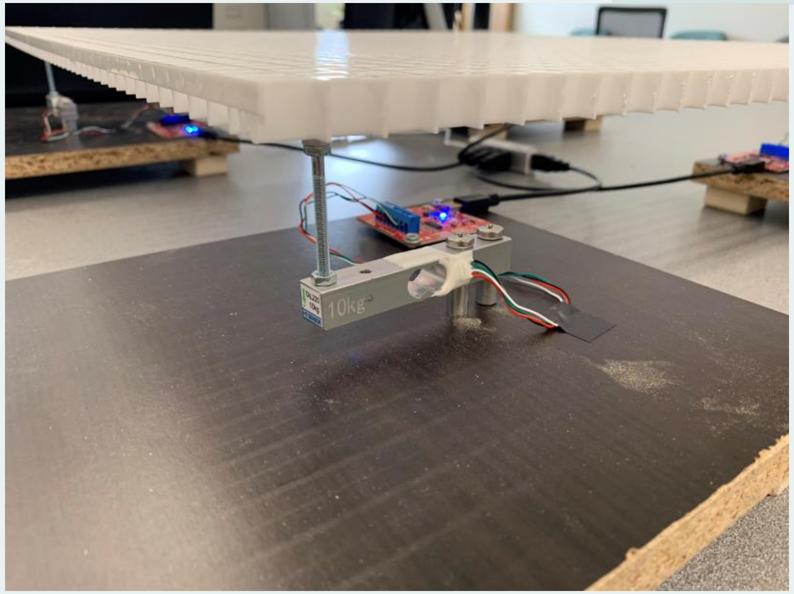
1. Is the drone overloaded with heavy payload?
2. Is the center of gravity aligned with the geometric center of the air frame?
3. Are all motors able to spin as directed by the flight controller?
4. Are all propellers intact and able to generate the expected thrust?
5. Does the flight compass need re-calibration?

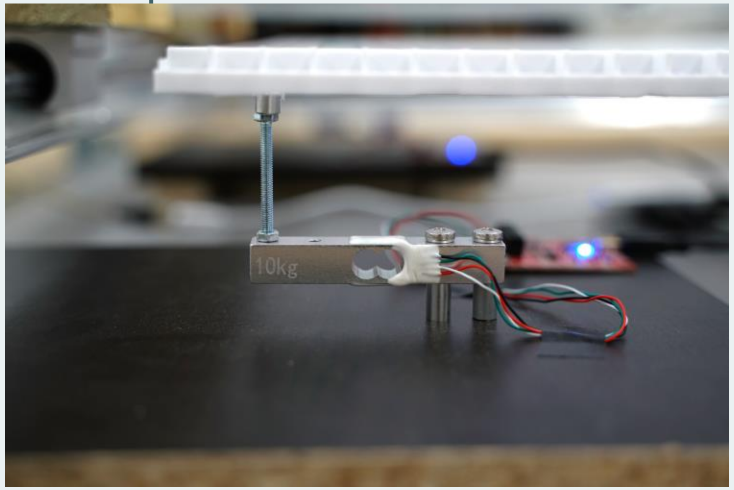
Can we perform these checks remotely?

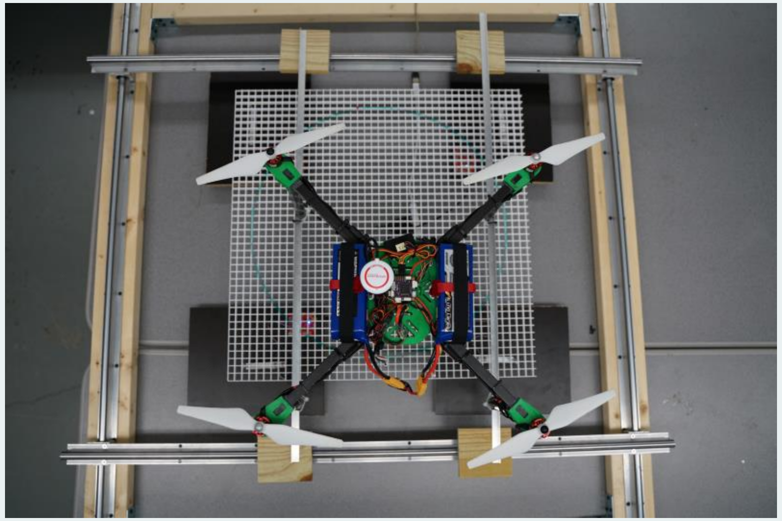
Construction of the landing platform for proof of concept

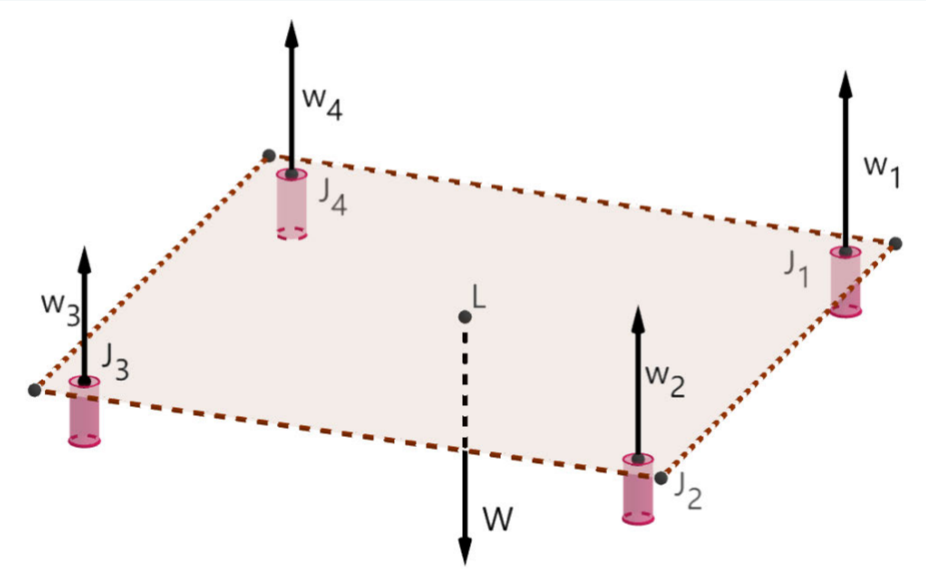


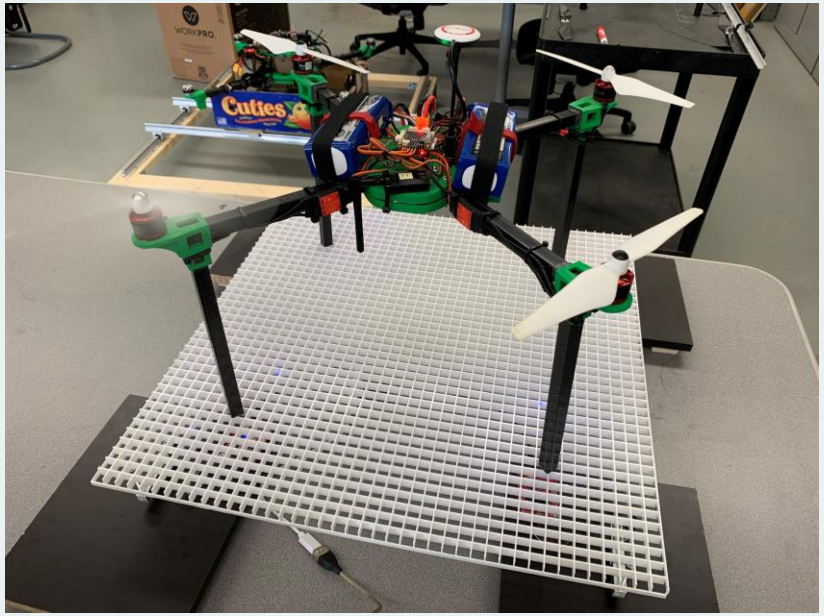












Measurement Procedure

Land still, read load cell measurements $w_j,\ j = 1, \dots, 4$.
Calculate:

1. Total weight $W = \sum_j w_j$
2. Center of gravity (projected on the platform plane)

Run a propeller motor, to create a new force F acting on the rigid body at the propeller center (x, y).
The change in weight will be sensed by the load cells, recorded as $\Delta w_j,\ j = 1, \dots, 4$.
Calculate F and (x, y) by equilibrium condition:

1. $F = \sum_j \Delta w_j$
2. $(x, y) = \left(\frac{1}{F}\sum_j \Delta w_j x_j^S,\ \frac{1}{F}\sum_j \Delta w_j y_j^S\right)$, where (x_j^S, y_j^S) is the location coordinate of load cell j.

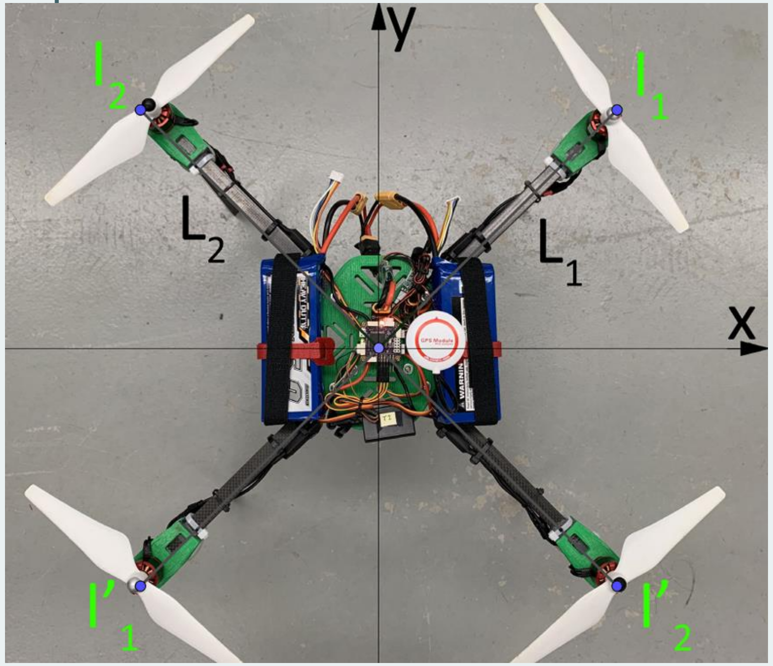
If F is smaller than the expected thrust, the propeller or motor is damaged.

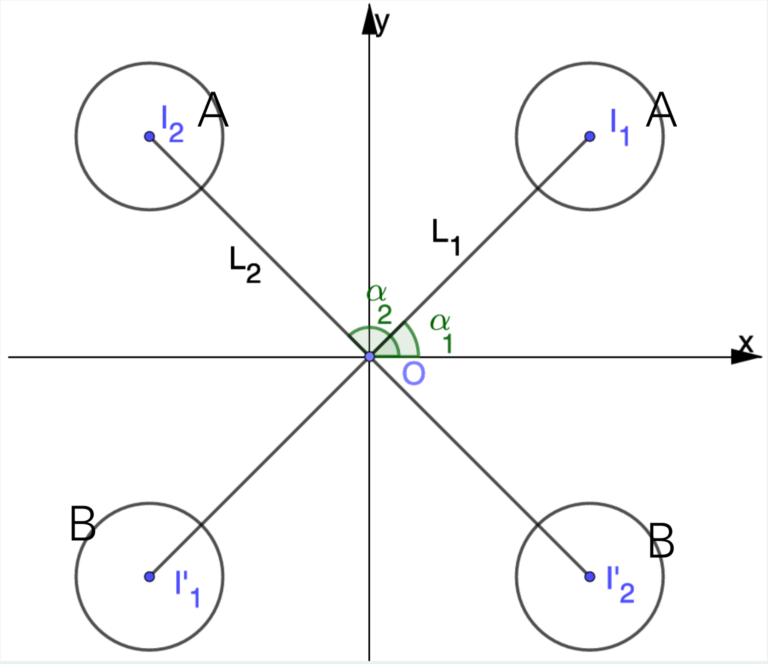
Inference by nonlinear regression

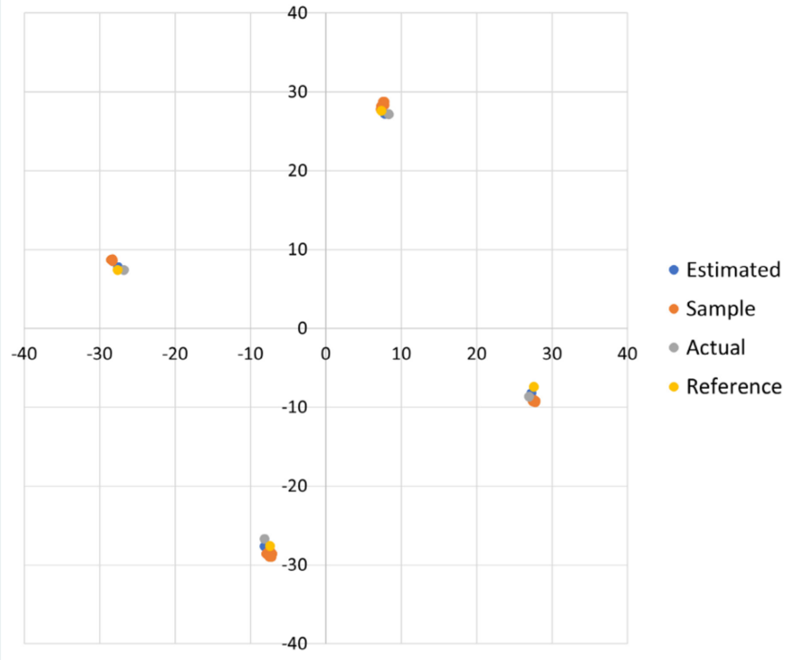
Record many samples of the force locations (x, y) for each propeller pair.
$(x_{i,s,k}, y_{i,s,k})$ is the s-th sample for propeller k in the i-th pair, $k \in \{A, B\}$, define $\operatorname{sgn} A \coloneqq 1$, $\operatorname{sgn} B \coloneqq -1$
L_i, α_i are the arm length and angle of propeller pair i, which are known by the airframe design

Infer the geometric center $(\hat{x}, \hat{y})$ and airframe orientation θ by solving the **nonlinear least squares**:

$$\text{Minimize}_{\hat{x},\hat{y},-\pi<\theta\le\pi} \sum_i \sum_s \sum_k \left(\left(\hat{x} + \operatorname{sgn}(k)\, L_i \cos(\alpha_i + \theta) - x_{i,s,k}\right)^2 + \left(\hat{y} + \operatorname{sgn}(k)\, L_i \sin(\alpha_i + \theta) - y_{i,s,k}\right)^2 \right)$$





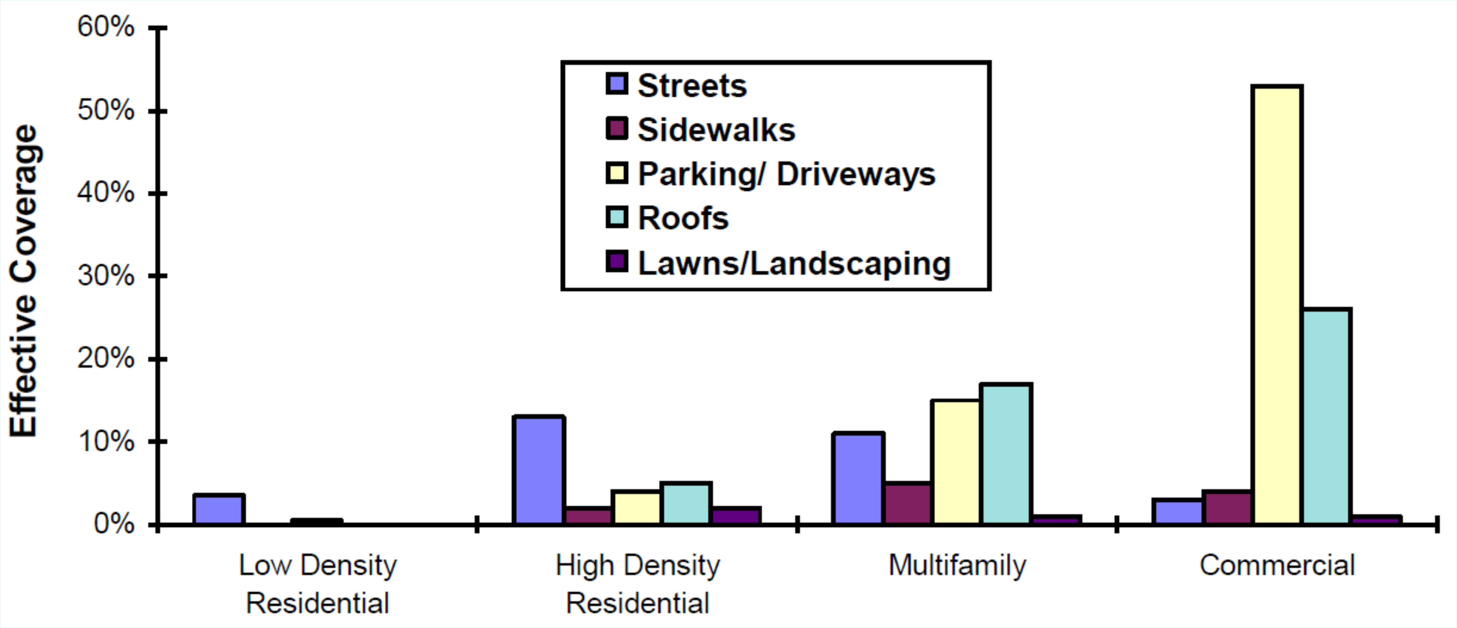


Z. Zhou and Y. Liu* (2021). A smart landing platform with data-driven analytic procedures for UAV preflight safety diagnosis, *IEEE Access*
US. Patent 63/249,752 (pending)

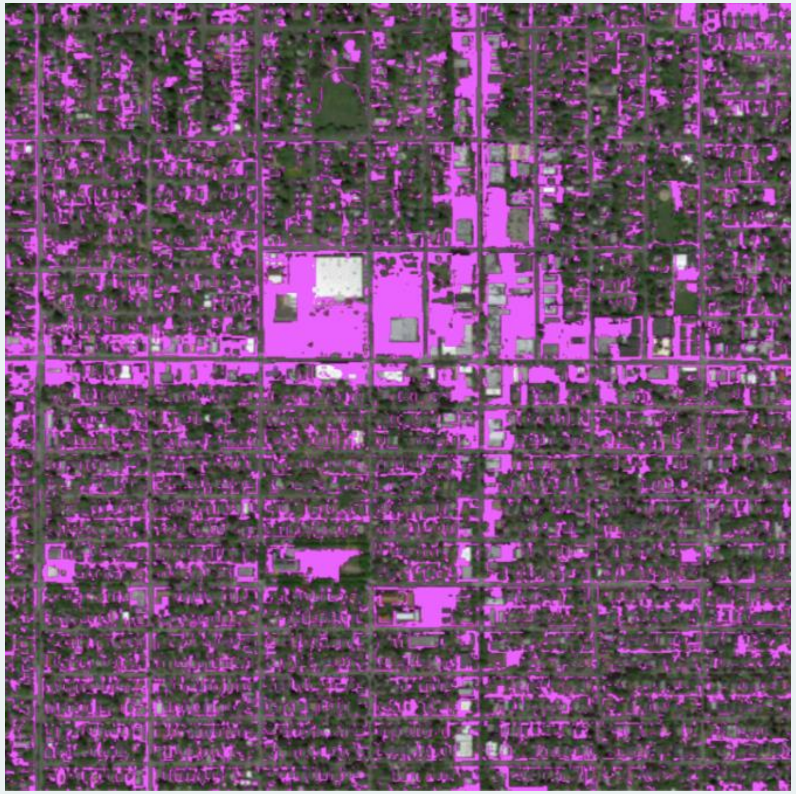

WORKPRO
Title: CG and Propeller Position
ARDUPILOT
A Decade of Cooking
CONTROL
QUALITY

Where should a drone land in emergency?

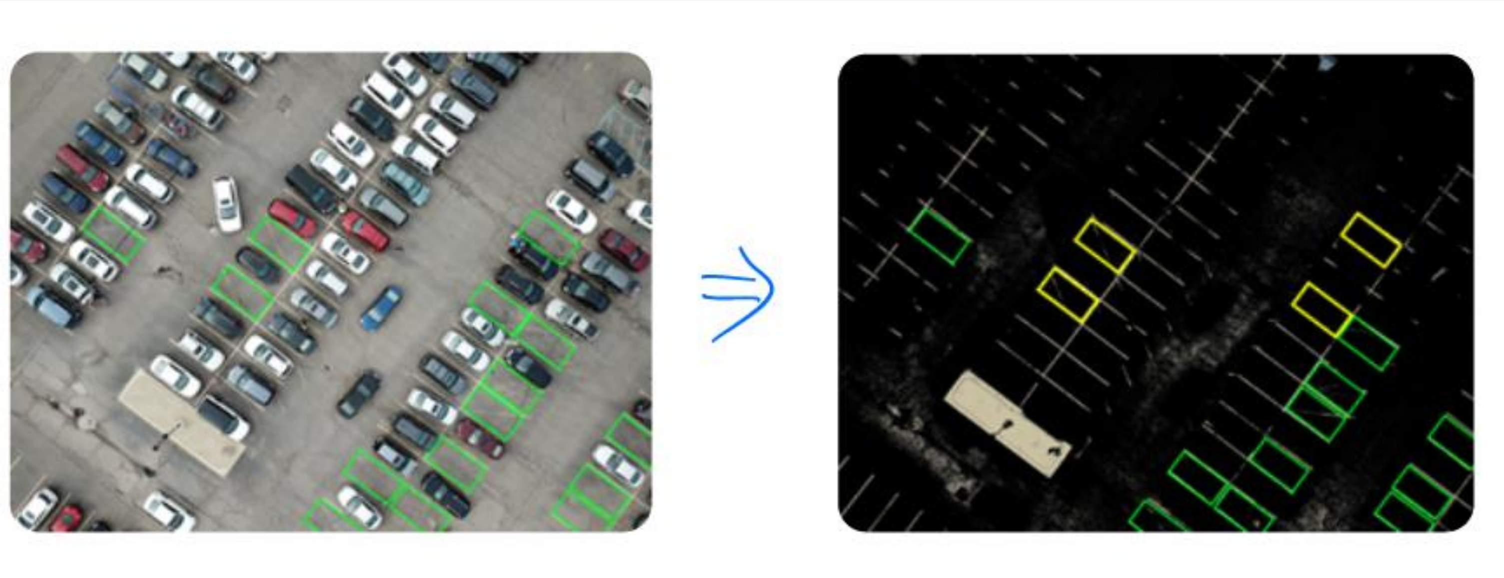
Parking spaces are ubiquitous in cities



Surface Coverage (Arnold and Gibbons, 1996)

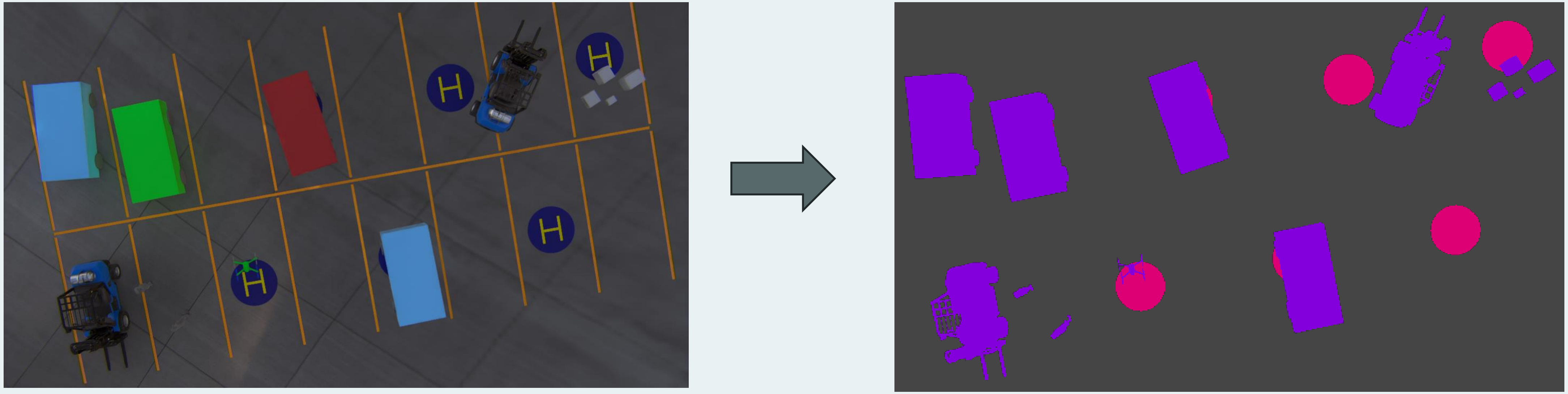


Surface parking landcover processed from multispectral imagery in Seattle (Eric Scharnhorst, 2018)

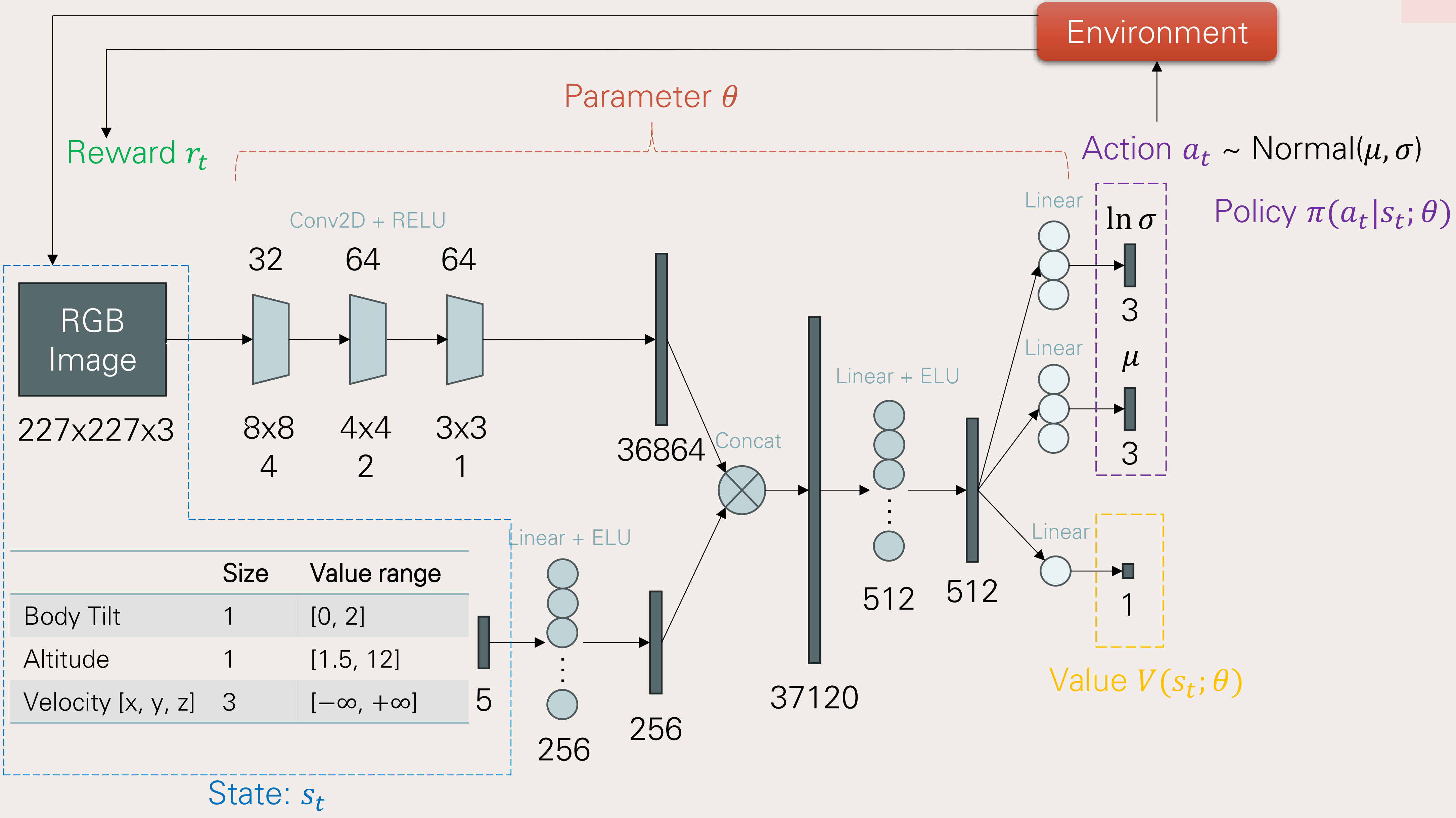


Train & validate an AI agent in a simulated environment

Deploy & test in a real drone



The agent's model



	Size	Value range
Body Tilt	1	[0, 2]
Altitude	1	[1.5, 12]
Velocity [x, y, z]	3	[$-\infty$, $+\infty$]

Advantage Actor-Critic (A2C) model

Actor's network learns the policy $\pi(a_t|s_t;\theta)$, the probability distribution of a_t conditioning on s_t

Critic's network learns the value $V(s_t;\theta)$, the expected return starting from s_t

Return $R_t := r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots$, the total discounted reward starting from time step t

Value $V(s_t;\theta)$: the expected return starting from state s_t, by following a policy parameterized by θ

The critic's network predicts the value of $V(s_t;\theta)$, denoted as $\hat{V}(s_t;\theta)$

$\hat{R}_t \coloneqq r_t + \gamma r_{t+1} + \cdots + \gamma^{T-1} r_{t+T-1} + \gamma^T \hat{V}(s_T;\theta)$ is the return obtained by taking actions $(a_t, a_{t+1}, \ldots, a_{t+T-1})$ in the next $T-1$ steps, then trusting the critic's predicted value for future steps.

$A_t \coloneqq \hat{R}_t - \hat{V}(s_t;\theta)$ is the *Advantage* of taking actions over trusting the critic's prediction all along

Loss function in model training

Actor's goal:

To increase the probability of choosing a_t when $A_t > 0$, and vice versa.

Policy Gradient (PG) method [1]:

$$\text{Maximize}_\theta \log \pi(a_t|s_t;\theta) \cdot A_t$$

Proximal Policy Optimization (PPO) method [2]:

$$\text{Maximize}_\theta \ \text{CLIP}(\phi_t(\theta), 1-\epsilon, 1+\epsilon) \cdot A_t, \text{ where } \phi_t(\theta) \coloneqq \frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta_{\text{old}})}$$

Critic's goal:

To increase prediction accuracy,

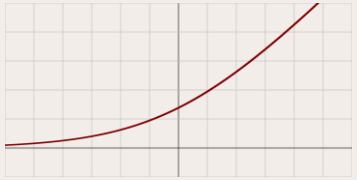
$$\text{Minimize}_\theta \ (V(s_t;\theta) - V(s_t;\theta_{\text{old}}) - A_t)^2$$

[1] Mnih et al. (2016) Asynchronous Methods for Deep Reinforcement Learning
[2] Shulman et al. (2017) Proximal Policy Optimization Algorithms

Loss function in model training

To ensure sufficient exploration, the action distribution should not converge too quickly to a (suboptimal) deterministic policy.

Two measures were taken:



1. Use 'softplus' activation function $y = \ln(1 + e^x)$ on the model output logstd (i.e., $\log \sigma$), so σ is always above 1.

2. Add a term to maximize the entropy [3] of the action distribution in the loss function, which exerts an upward push to σ.

To bound the magnitude of μ (the mean of action), minimize

$$\text{b\_loss} := [(\mu - 1.1)^+]^2 + [(\mu + 1.1)^-]^2$$

[3] Williams and Peng (1991), *Function optimization using connectionist reinforcement learning algorithms.*

Model training

The overall loss function used in PPO:

$$\text{Loss} = -\text{CLIP}(\phi_t(\theta), 1-\epsilon, 1+\epsilon) \cdot A_t + (V(s_t;\theta) - V(s_t;\theta_{\text{old}}) - A_t)^2 - 0.01 \cdot \text{entropy} + 0.0001 \cdot \text{b\_loss}$$

<u>In each training epoch:</u>

For actor = 1, …, N do

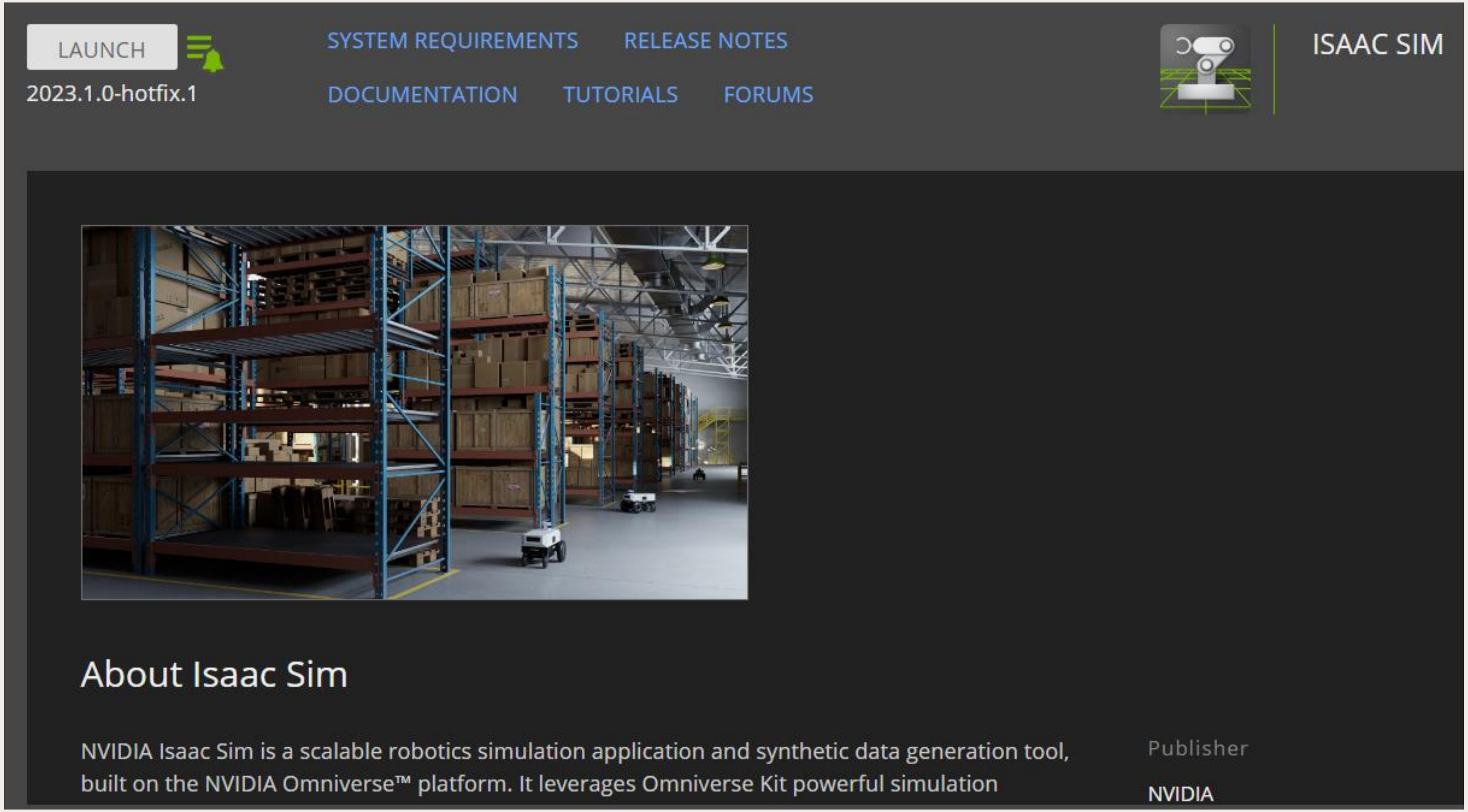
Run policy π_{old} in environment for T timesteps

Compute advantages $A_t, \dots, A_{t+T}$

Minimize Loss wrt θ, for K mini-epochs with minibatch size M

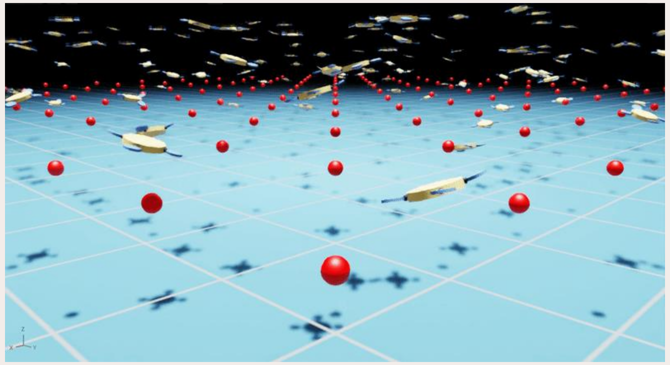
$\theta_{\text{old}} \leftarrow \theta$

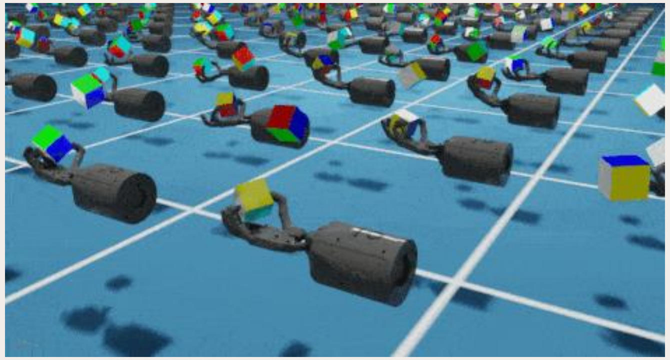
NVIDIA Omniverse and Isaac Sim



Isaac Gym

GPU-accelerated vectorized RL training environments featuring physics-based simulation, photo-realistic rendering, domain randomization and sim-to-real support (e.g., via ROS).



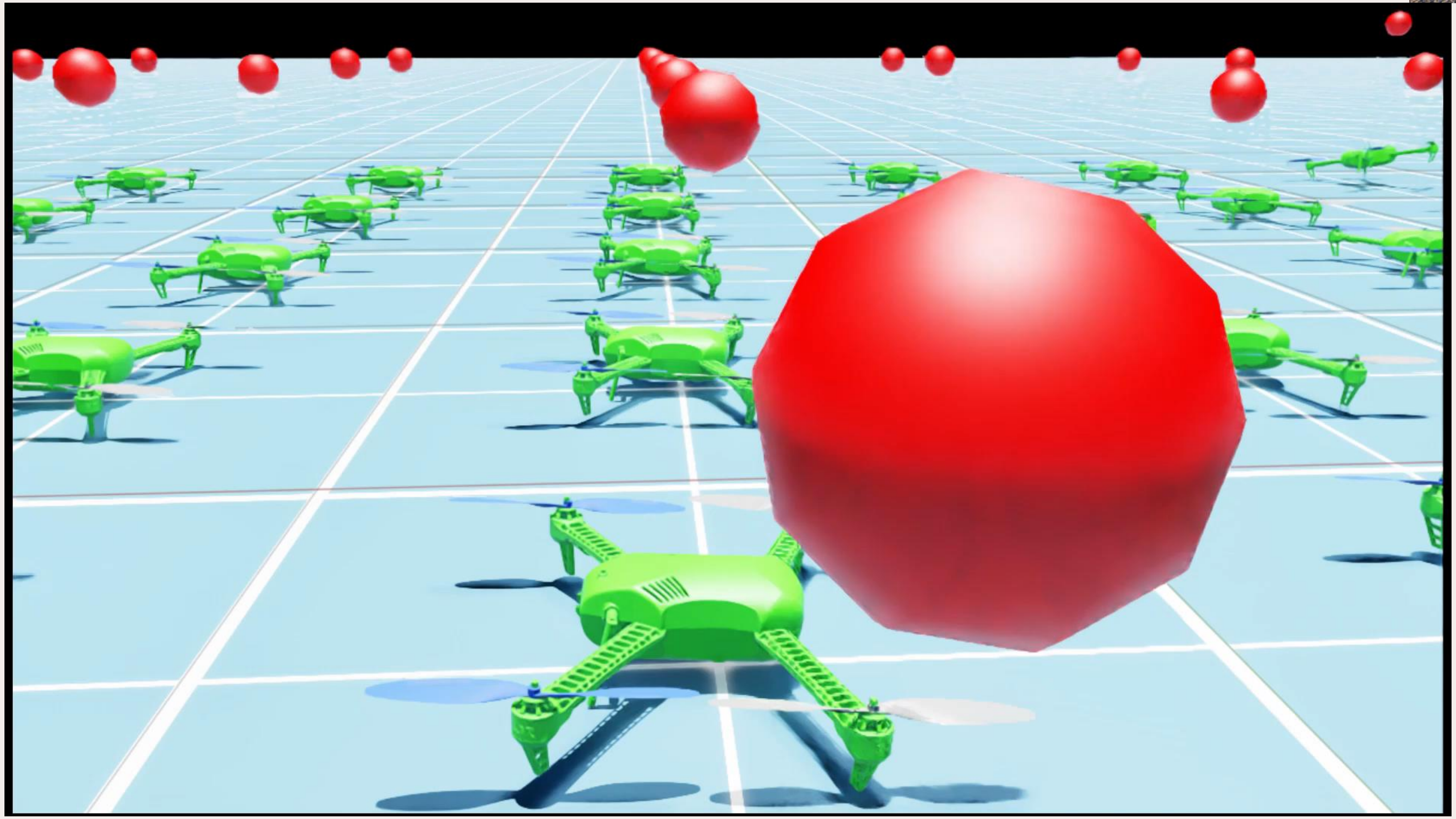


Relevant programs in [Isaac Gym](), such as the Quadcopter and Crazyflie tasks, have some limitations:

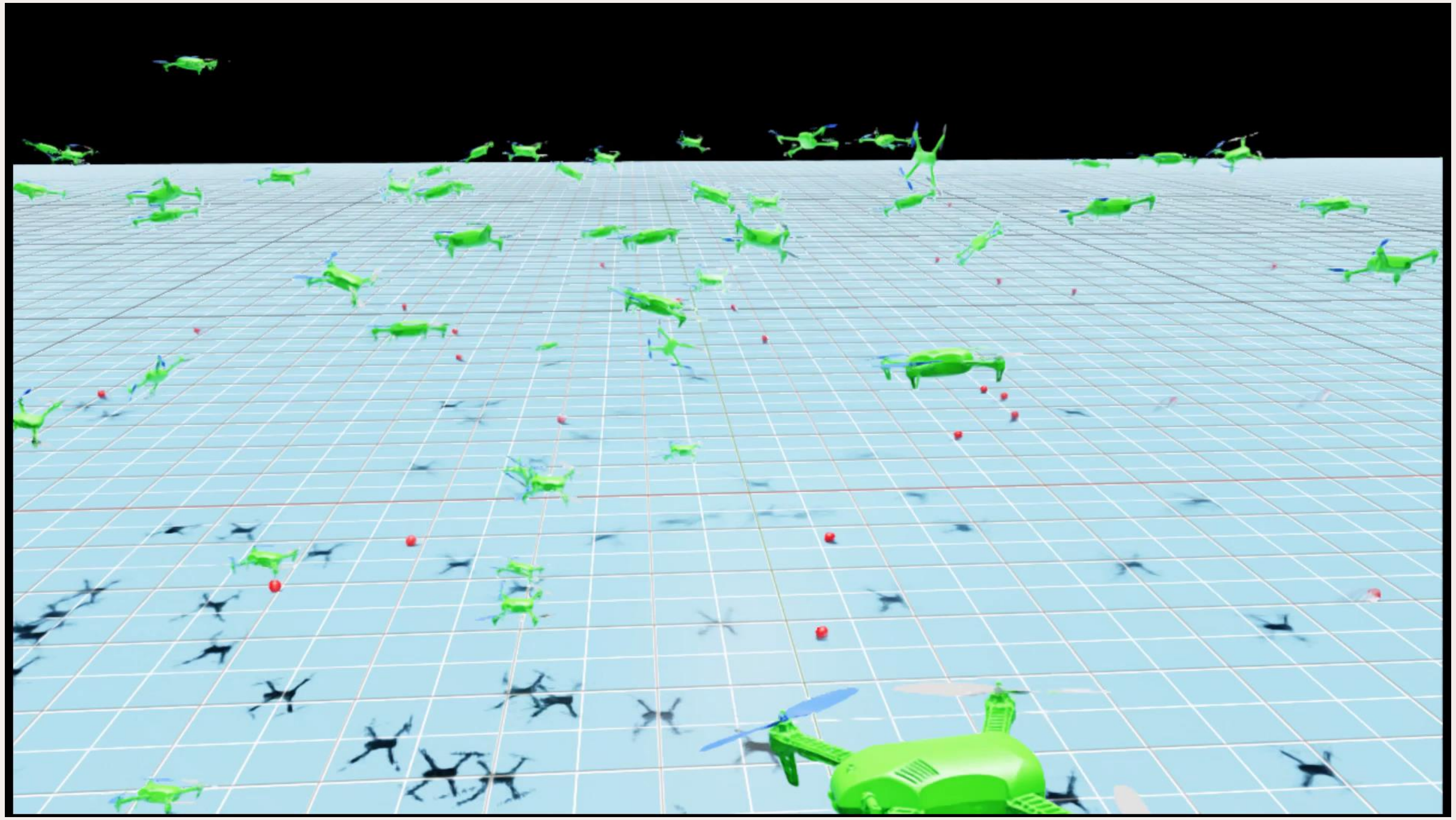
1. Actions to be learned are low-level control targets, such as forces and thrusts with high-frequency updates, i.e., dt = 0.01 s
2. Observations are uni-modal, i.e., IMU inputs
3. Not suited for complex task configurations, e.g., hierarchical decisions, task switching, etc.

The Drone

A digital model of the 3DR's Iris quadrotor, with a downward-facing camera on a stabilized gimbal.



The flight controller

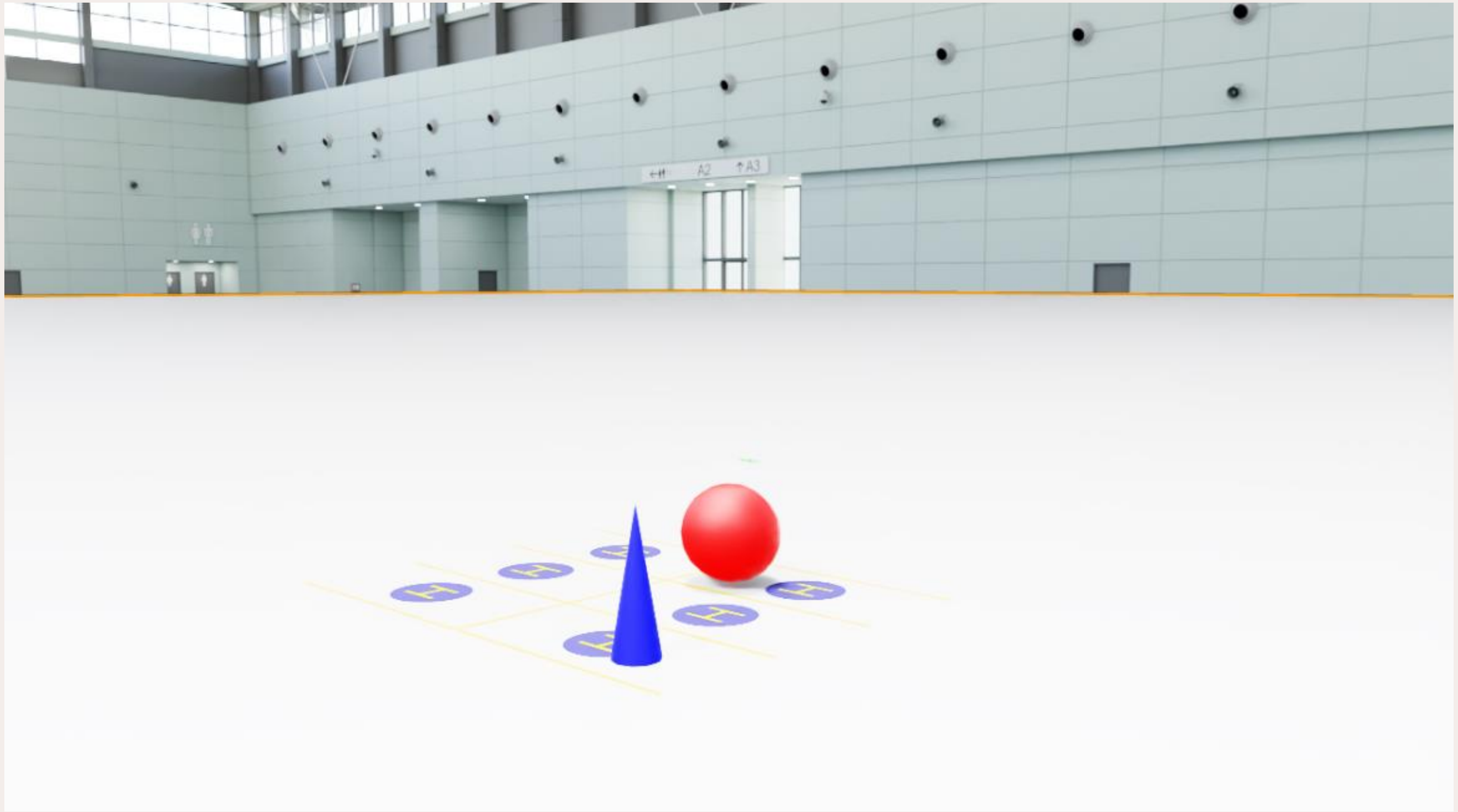


The RL model decides where to go, i.e., target coordinate [x, y, z] in body frame, for the next few physical steps.

The physical execution, i.e., the thrusts and torque generated by the propellers, is handled by the FC.

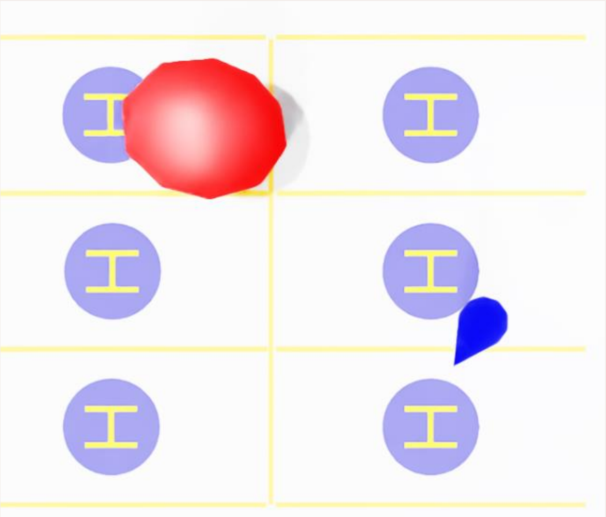
I implemented a **tensorized version** of the nonlinear controller presented in Daniel Mellinger and Vijay Kumar, *Minimum snap trajectory generation and control for quadrotors*, 2011, so that **many drones can be controlled in parallel efficiently via GPU**.

The landing scene



Target: six available parking spaces

Obstacles: a cone and a rolling ball



Reward Function

DNT = Horizontal Distance to Nearest Target
DNO = Horizontal Distance to Nearest Obstacle

Reward r_t = Target + Up + Altitude – 3 + Done*(OnTarget – Obstacle)

Target = 1 / (1 + DNT)

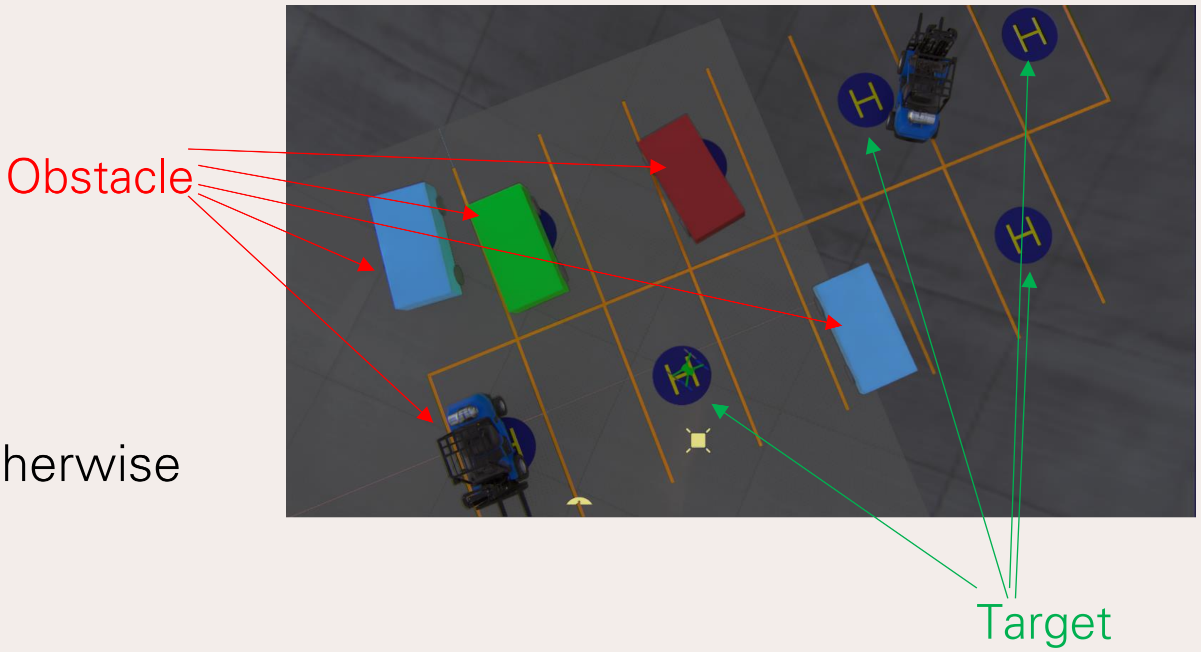
Up = 1 / (1 + 10*Tilt)

Altitude = 1 / (1 + |BodyPosZ – 1.5|)

Done = 300 if BodyPosZ < 1.5; 0 otherwise

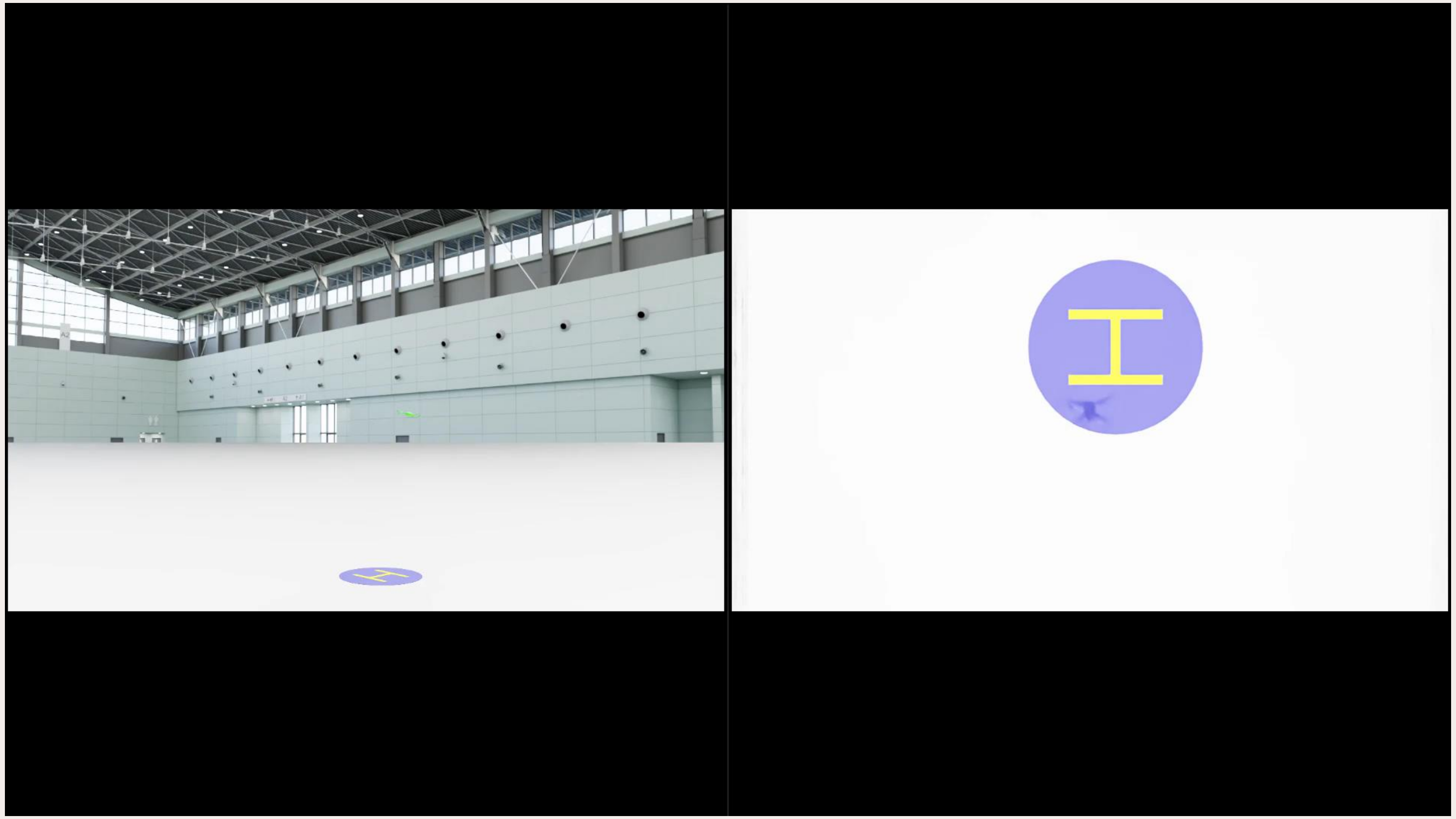
OnTarget = -4*DNT^2 + 1 if DNT < 0.5; 0 otherwise

Obstacle = exp(-2*DNO)



An episode ends if Tilt > 0.5 or BodyPosZ < 1.5 or BodyPosZ > 30 or Step Count > 400

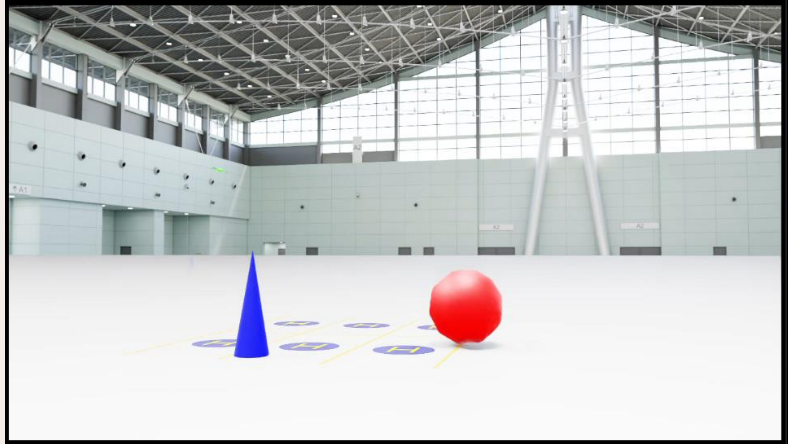
Simpler scene test result

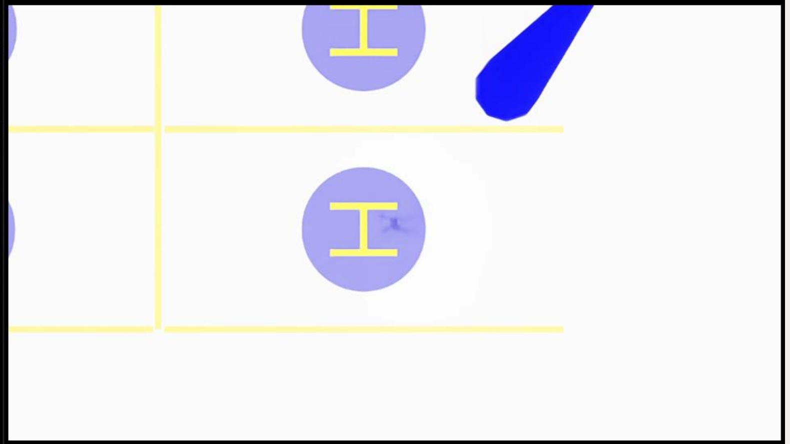


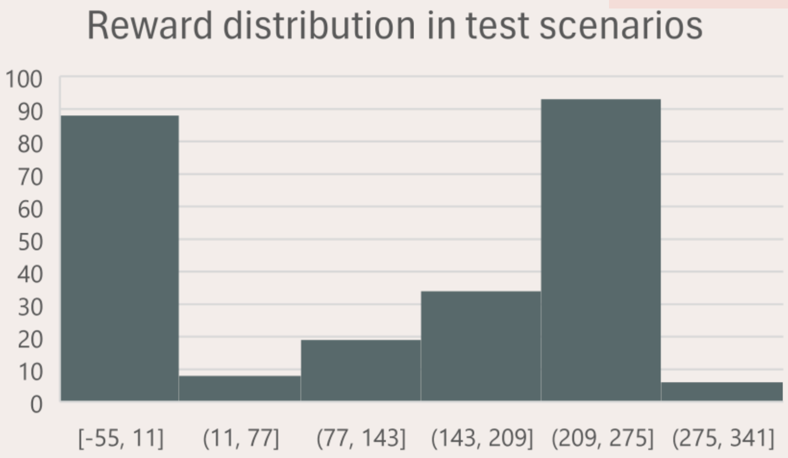
Single target, no obstacle

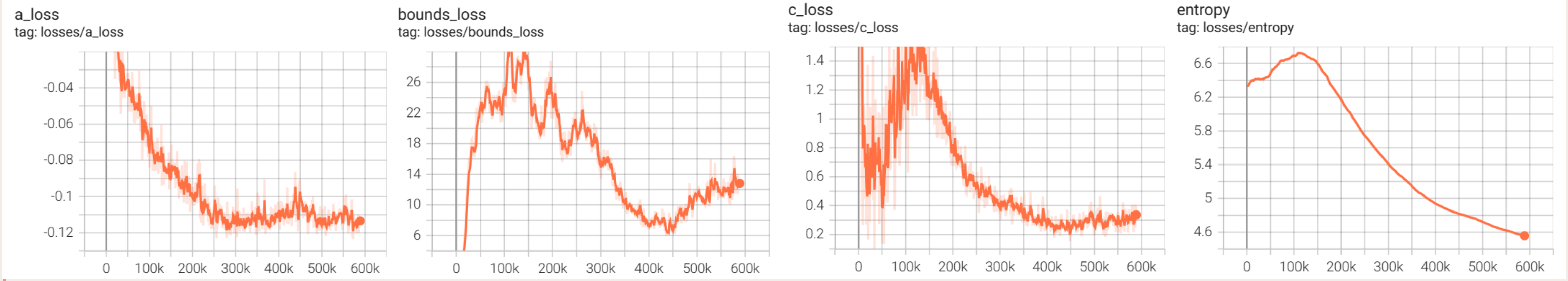
The trained model works well.

Multiple targets, multiple obstacles









In training: N=16, T=64, M=256, K=8. Adam Optimizer with adaptive learning rate starting at 0.001 was used.

Trained for 576 epochs, ~4.5 hr

Future Work

- Nonlinear optimization algorithms
- Sim-to-Real transfer of AI models
- Search and rescue, public safety, agriculture applications

Thank you

Yanchao Liu

yanchaoliu@wayne.edu