AirSim: A Powerful Simulator for Robotics Research

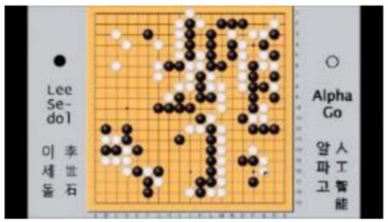
Yulai Weng
CIS 700/002: Topics in Safe Autonomy
Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania

02/12/2019

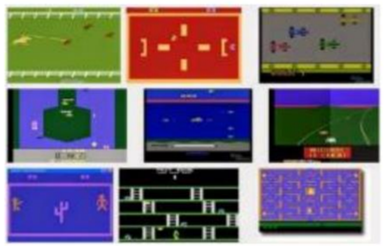
Can we use such ML methods to build systems that operate in real-world?

Successes of Machine Intelligence

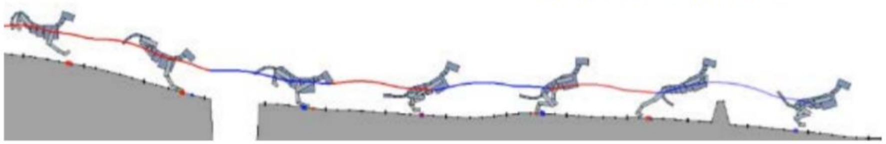
Real World Flying Systems



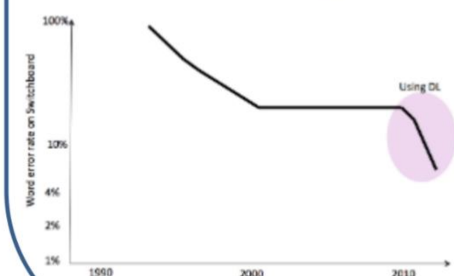
Silver et al. 2016

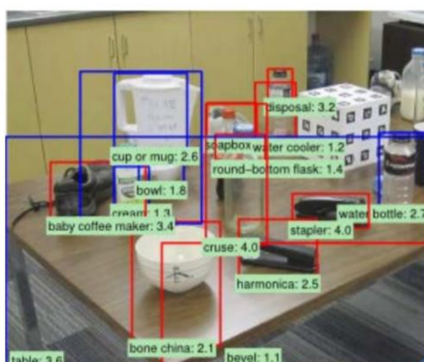


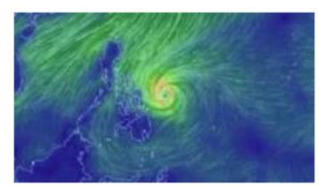
Mnih et al. 2013



Peng et al. 2016







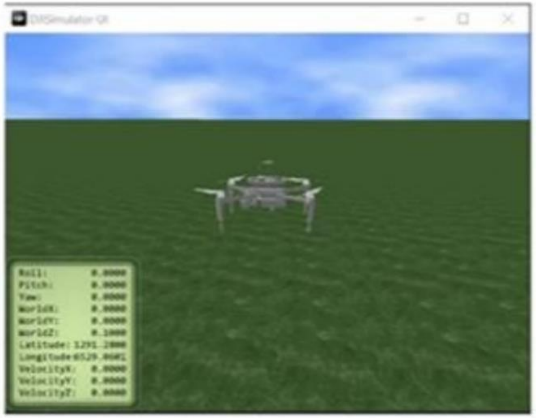
Three Fundamental Challenges

Lack of large amount of data	Computational Constraints	Operating in the Open World
• *High Sample Complexity of ML Methods*	• *Real-time Performance, Limited Memory and Compute Power*	• *Safety, Uncertainty, Mixed-Initiative Autonomy*

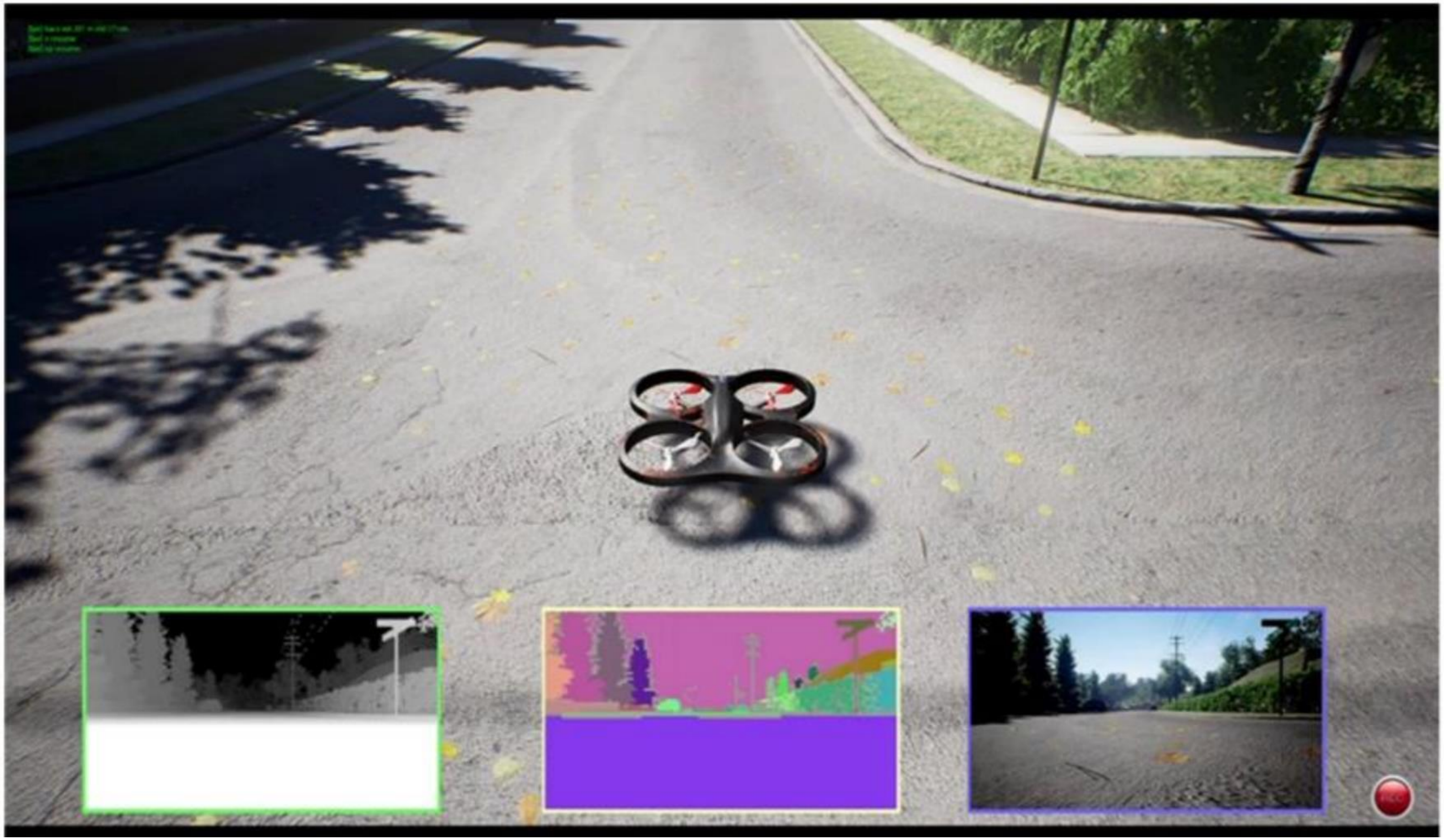
Penn Engineering

PRECISE
PENN RESEARCH IN EMBEDDED COMPUTING AND INTEGRATED SYSTEMS ENGINEERING

Old Version of UAV Simulators



AirSim: The Next Generation Simulator



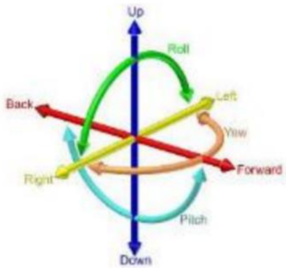
What Does Simulator Enables?

Generate lots of training data	• Wide variety of environment, day of time, weather patterns • No legal hassle, much cheaper and safe
Develop autonomy algorithms	• Run exact same code that would be run onboard • Slow down or accelerate simulated time
Test perception algorithms	• If it doesn't work in simulator then it won't work in real world • Use sensors with varies parameters
Reinforcement Learning techniques	• Can fail thousands of time to learn patterns • Run in cloud for distributed learning

Why Use Unreal Engine?

Actual footage captured from simulated drone in AirSim with Open World Environment

The Heart of the Vehicle

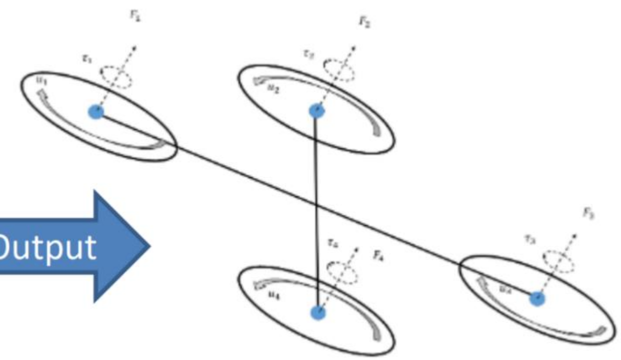


Sensor Input

Sensor Input

Sensors generate information about the world around

Motor Output

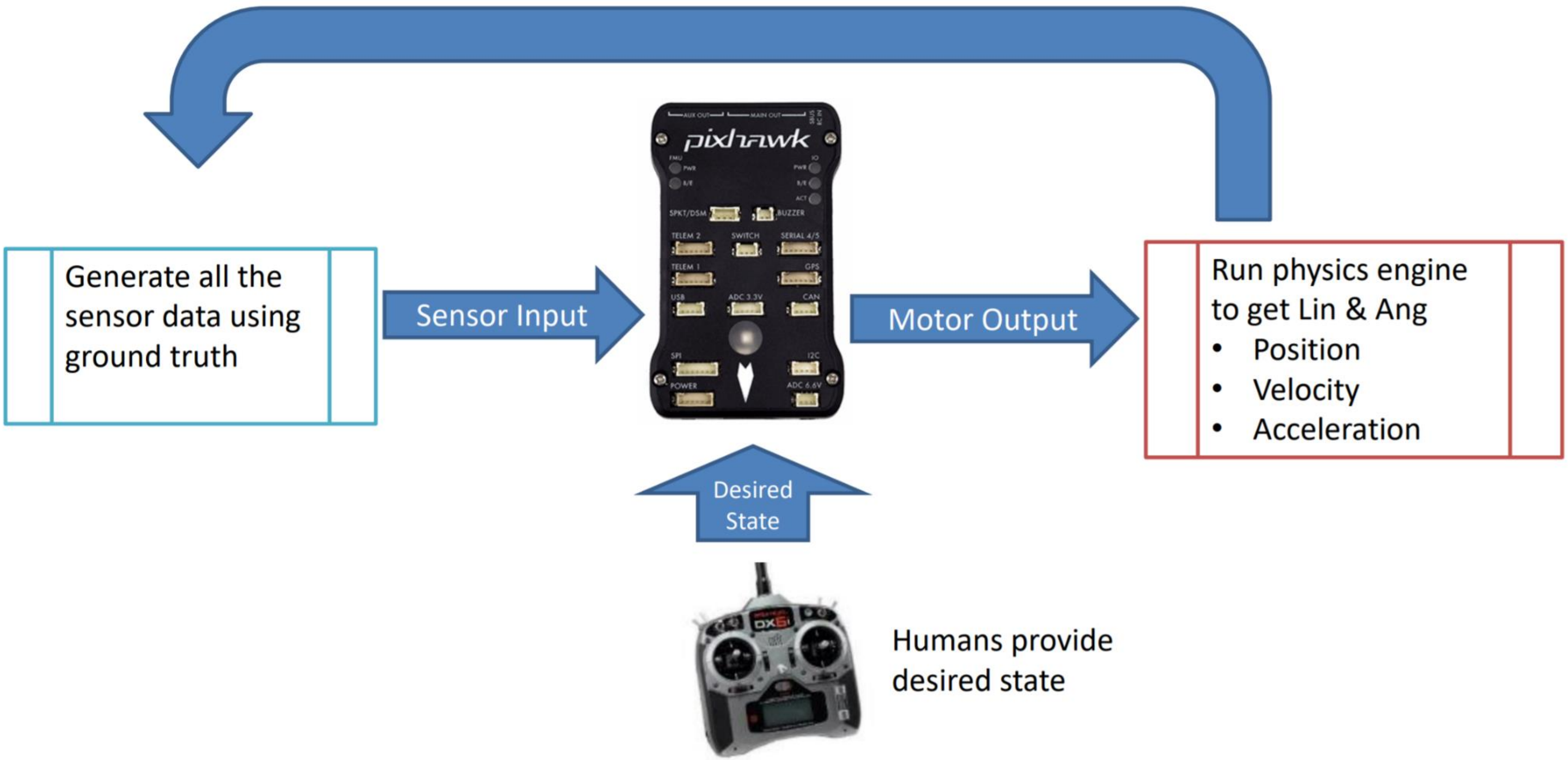


Motor voltages drives the propeller

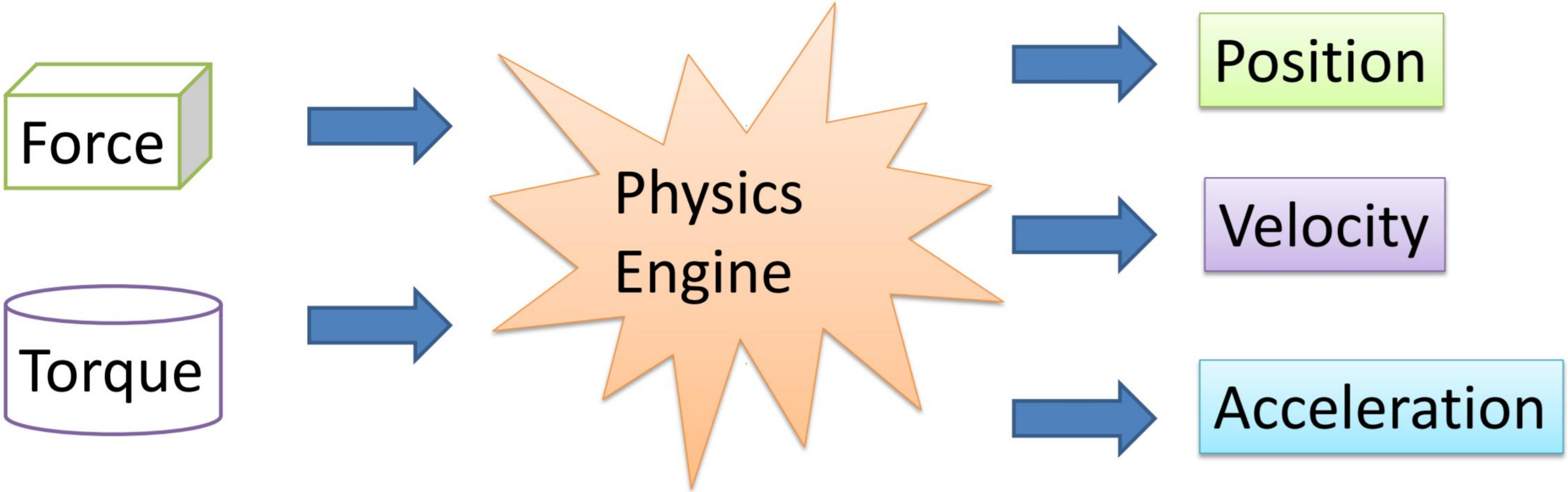
Desired State

Humans provide desired state

How Simulator Works?

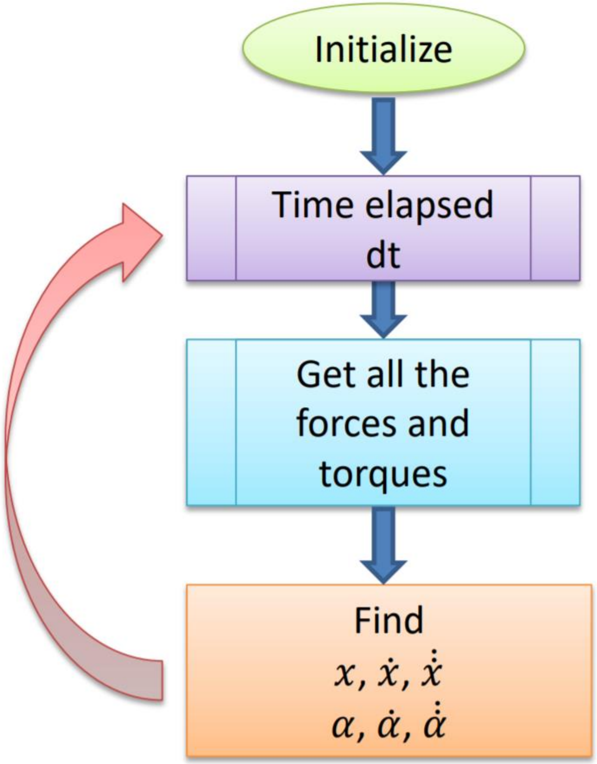


What Does Physics Engine Do?

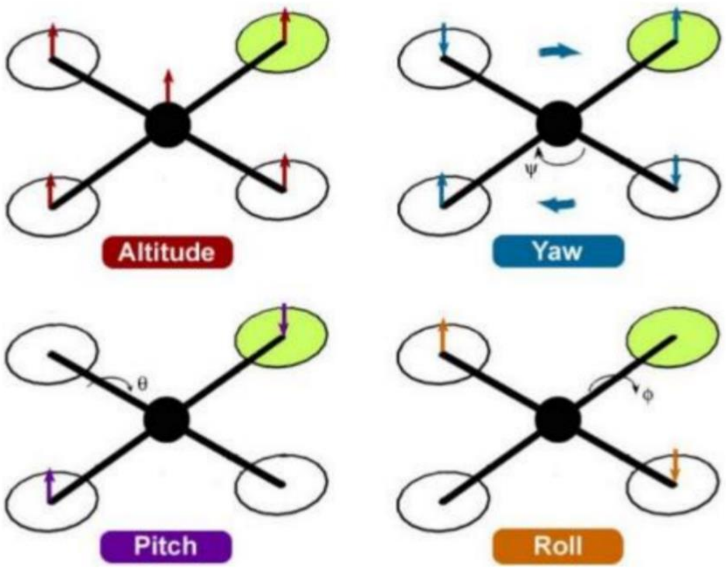


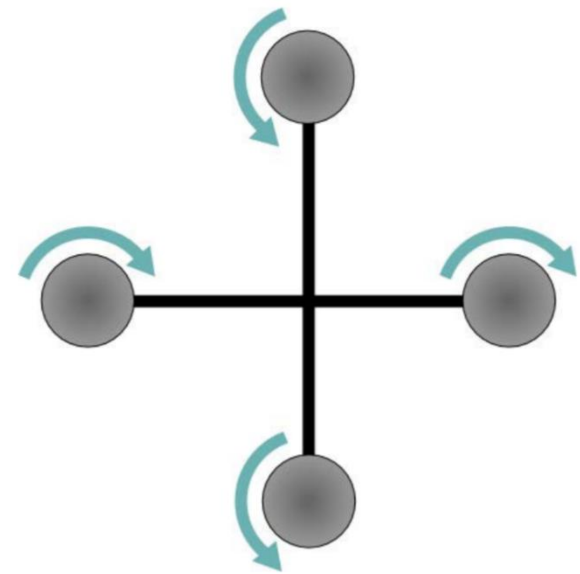
Linear & Angular Flavors
Total of 3 + 3 = 6 vectors

The Physics Loop



May the Force & Torque be with You





$$F = MotorSignal * K_F \qquad K_F \propto \rho D^4$$

$$T = MotorSignal * K_T \qquad K_T \propto \rho D^5$$

Physics: Dynamics

$$acceleration_{next} = \frac{force}{mass}$$

$$acceleration_{next} = \frac{torque\ - \omega \times (I\ \omega)}{I}$$

$$velocity_{next} \mathrel{+}= acceleration_{next}\ * dt$$

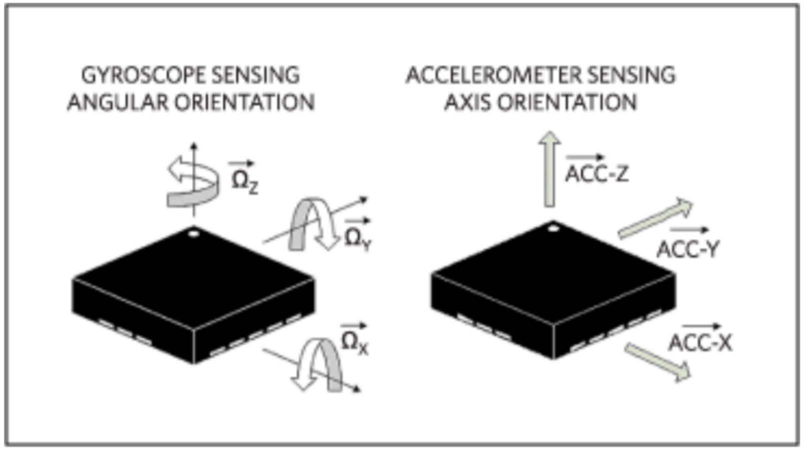
$$\omega_{next} \mathrel{+}= acceleration_{next}\ * dt$$

$$position_{next} \mathrel{+}= velocity_{next}\ * dt$$

$$q_{next} = q_{prev}\ * Q(\omega\ * dt)$$

Simulating IMU

$$IMU = Gyroscope + Accelerometer$$



Physics engine tells us,

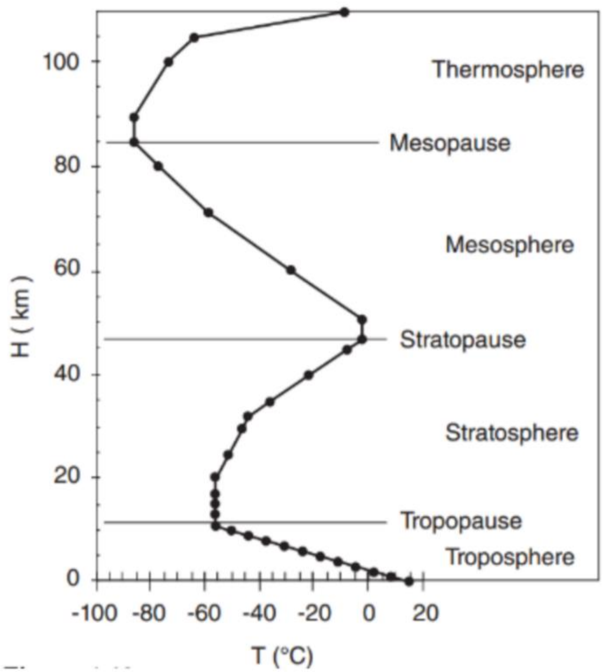
- Angular velocity
- Linear acceleration

So just add noise, bias and drift!

$$\omega^{\text{out}} = \omega + \eta_a + b_t, \qquad \text{where } \eta_a \sim N(0, r_a) \text{ and}$$

$$b_t = b_{t-1} + \eta_b, \qquad \text{where } \eta_b \sim N(0, b_0\sqrt{\frac{dt}{t_a}}).$$

Simulating Barometer



$$P = (101.325\text{kPa})\cdot(288.15\text{K}/T)^{-5.255877} \qquad H \le 11\text{ km}$$

$$P = (22.632\text{kPa})\cdot\exp[-0.1577\cdot(H-11\text{ km})] \qquad 11 \le H \le 20\text{ km}$$

$$P = (5.4749\text{kPa})\cdot(216.65\text{K}/T)^{34.16319} \qquad 20 \le H \le 32\text{ km}$$

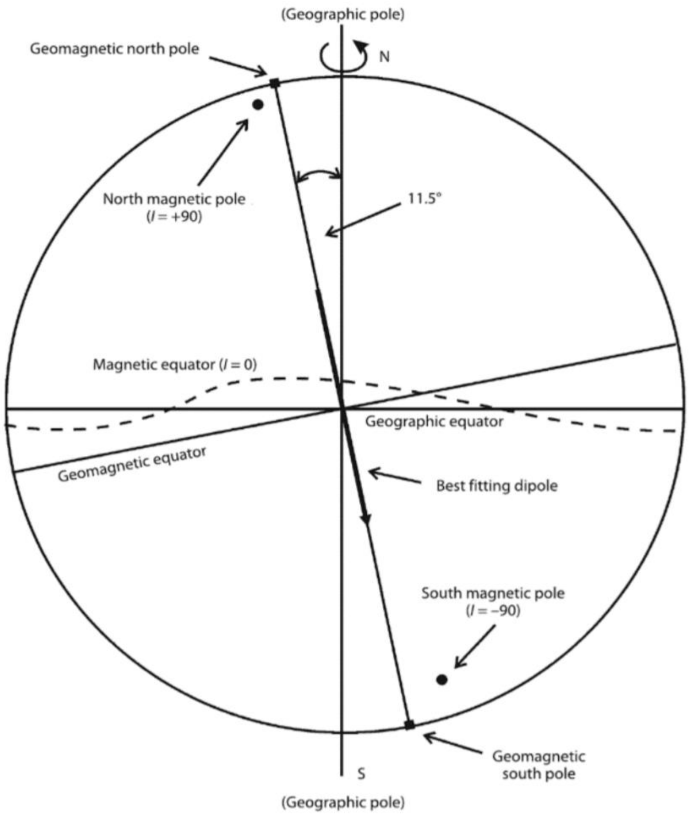
$$P = (0.868\text{kPa})\cdot(228.65\text{K}/T)^{12.2011} \qquad 32 \le H \le 47\text{ km}$$

$$P = (0.1109\text{kPa})\cdot\exp[-0.1262\cdot(H-47\text{ km})] \qquad 47 \le H \le 51\text{ km}$$

Density, ρ (kg/m^3) = $P/(RT_M)$
Speed of sound, C (m/s) = $(\gamma R T_M)^{1/2}$

Specific gas constant, R = 287.053 J/kg-K
Specific heat ratio, γ = 1.400

Simulating Magnetometer



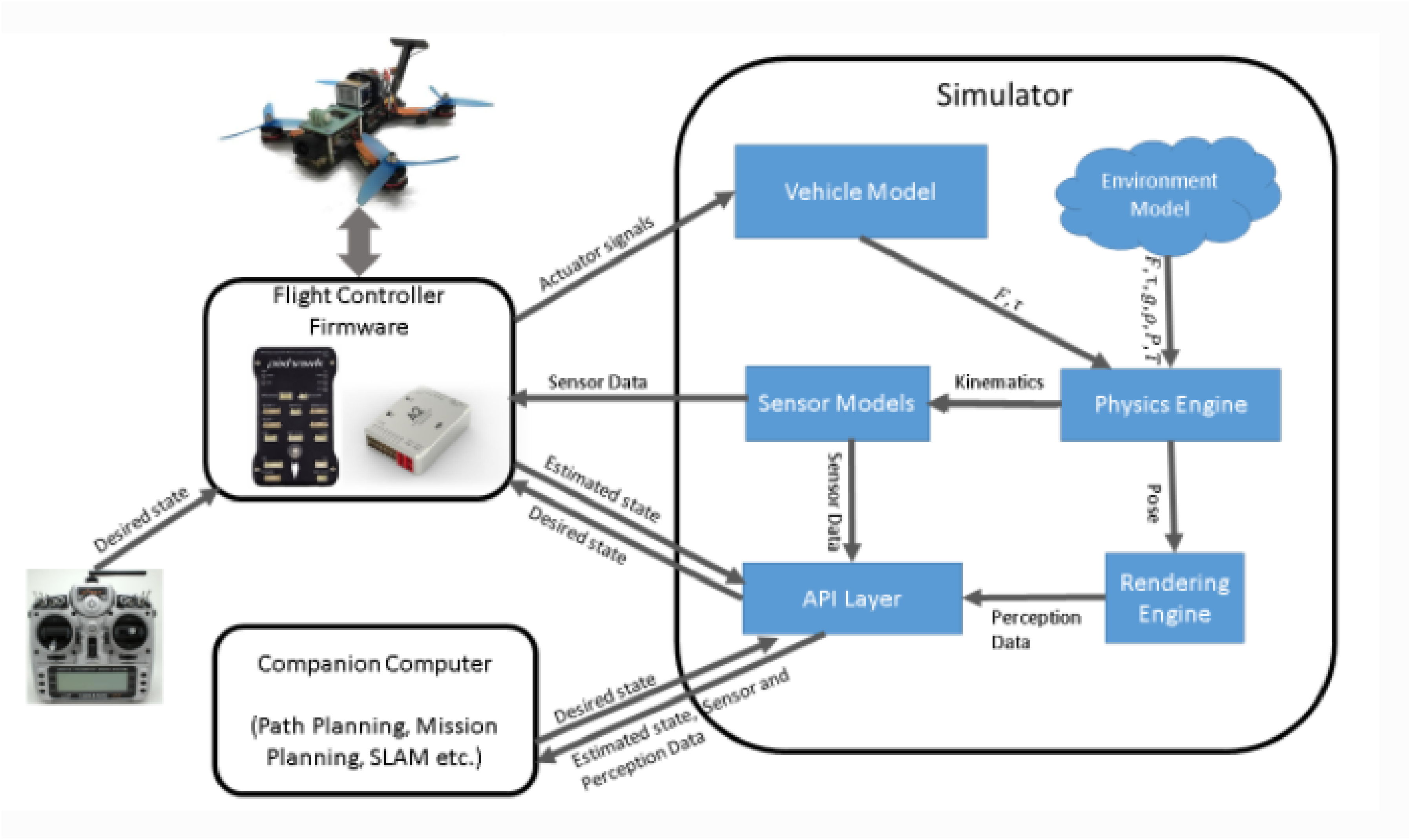
$$\cos\theta_m = \cos\theta\cos\theta^0 + \sin\theta\sin\theta^0\cos(\phi - \phi^0).$$

$$|B| = B_0\left(\frac{R_e}{R_e + h}\right)^3\sqrt{1 + 3\cos^2\theta_m}$$

$$\tan\alpha = 2\cot\theta_m \quad \text{and} \quad \sin\beta = \begin{cases} \sin(\phi - \phi^0)\frac{\cos\theta^0}{\cos\theta_m}, & \text{if } \cos\theta_m > \sin\theta^0\sin\theta \\ \cos(\phi - \phi^0)\frac{\cos\theta^0}{\cos\theta_m}, & \text{otherwise.} \end{cases}$$

$$H = |B|\cos\alpha \qquad Z = |B|\sin\alpha$$
$$X = H\cos\beta \qquad Y = H\sin\beta.$$

The Architecture of AirSim



AirSim is Open Source

Unreal Engine 4

- Designed as UE4 plugin
- Just drop in to 100s of realistic environments

APIs for Dozens of Languages

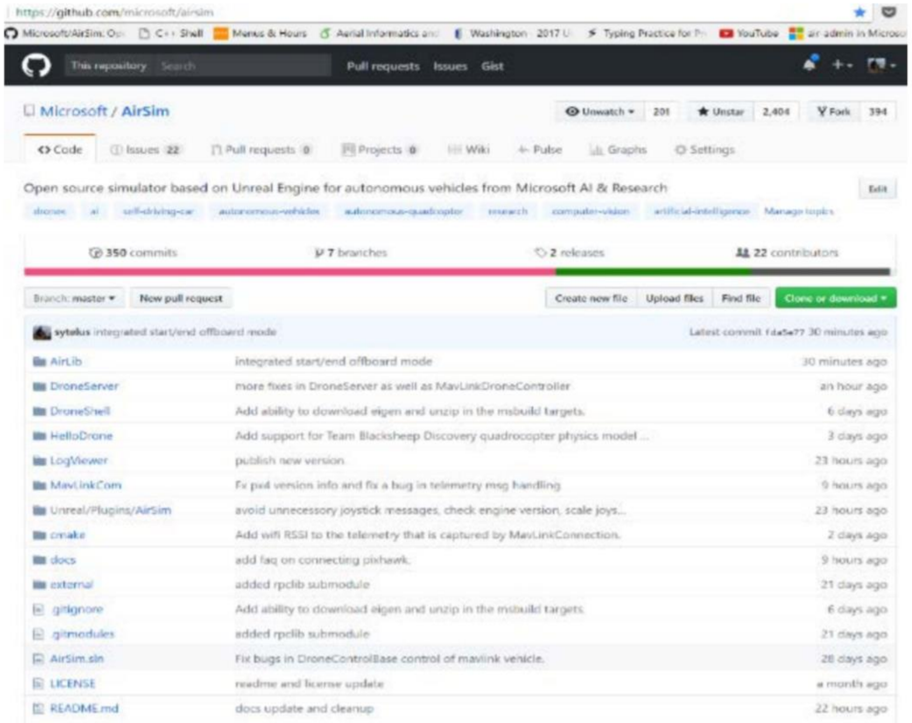
- Get camera images
- Send commands to the vehicle

C++ Header-Only Library

- Eigen library as only dependency
- Cross-platform

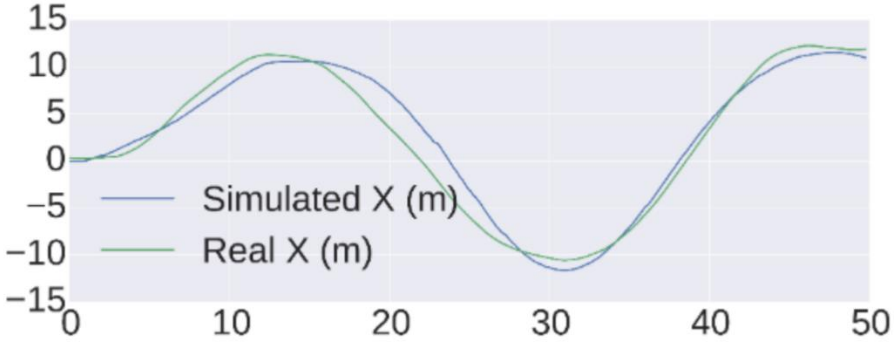
Highly Extensible

- Add new vehicles models
- Add new modes



Experiment: Simulation vs. Reality

Simulated trajectory in purple, real trajectory in red



AirSim Has APIs

```
from PythonClient import *
import cv2
import sys

client = AirSimClient('127.0.0.1')

# get depth image
result = client.setImageTypeForCamera(0, AirSimImageType.Scene)

# show image in opencv
rawImage = np.fromstring(result, np.int8)
png = cv2.imdecode(rawImage, cv2.IMREAD_UNCHANGED)
cv2.imshow("Camera Image", png)
```

Few lines of Python code can get you FPV image from drone!

Scene from Unreal Boy with a Kite environment

Make Drone Move in AirSim Using APIs

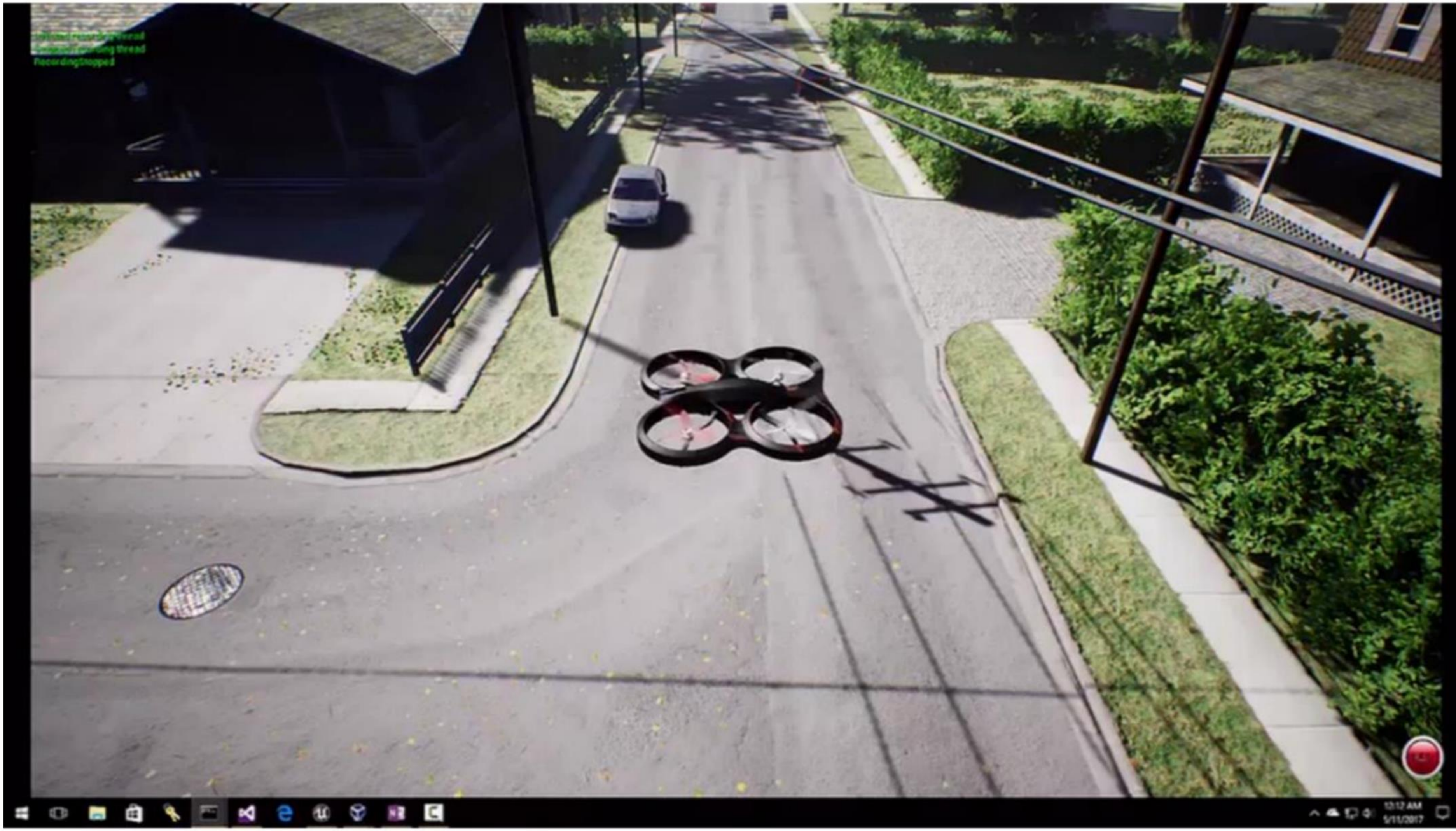
```
from PythonClient import *
import sys

client = AirSimClient('127.0.0.1')

# Stay 5 meters above ground
z = -5

# Fly!
client.moveOnPath([(0,-253,z),(125,-253,z),(125,0,z),(0,0,z)],
15, 0, DrivetrainType.ForwardOnly,
YawMode(False,0), 20, 1)
```

Same code can be ran from offboard computer on real drone!
Other languages available: C++, C#, Java and many more!

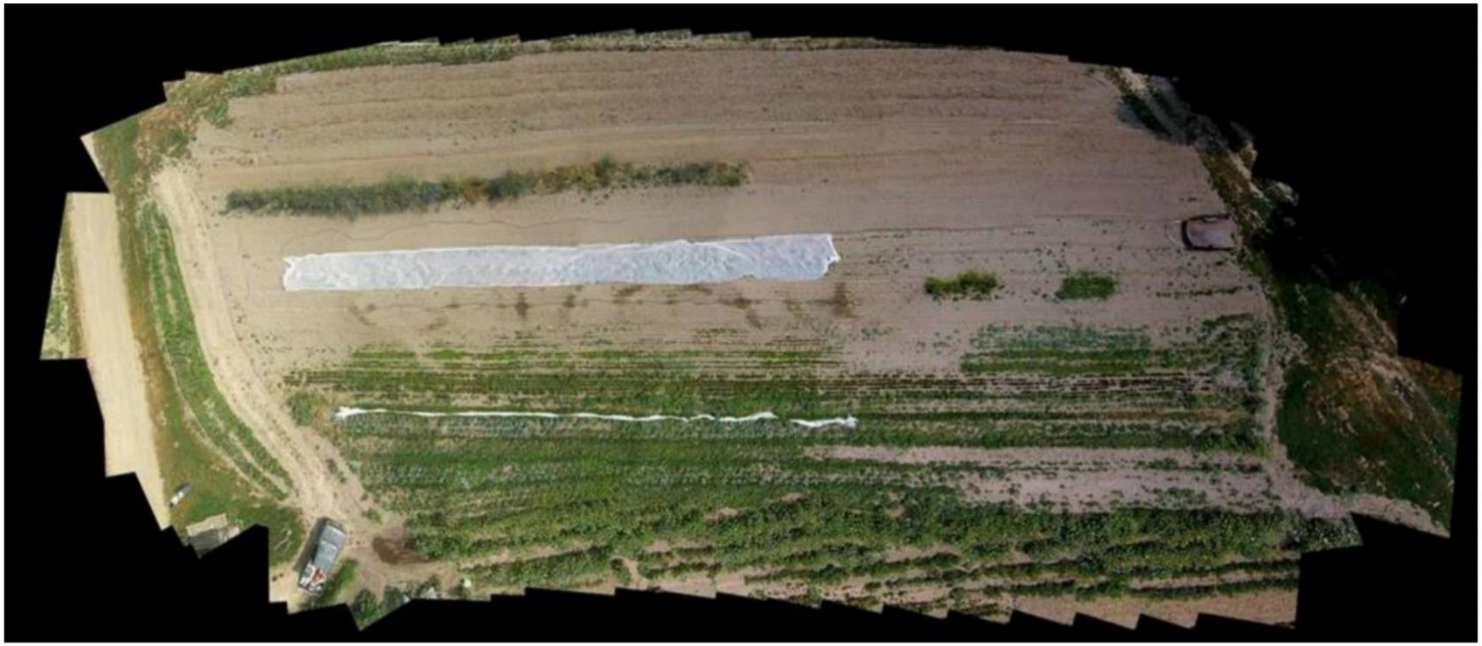


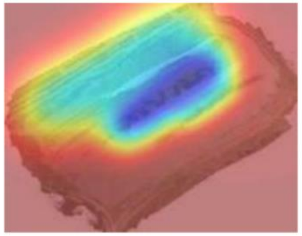
AirSim Extensibility

- New Vehicle Types
  - Car
  - Apollo Moon Lander
- New Sensor Types
  - Ultrasonics
  - LIDAR
- New Physics Engines
  - PhysX
  - Bullet
- New Sim Modes
  - Computer Vision
  - Just Physics

You can contribute on GitHub: https://github.com/microsoft/airsim

AgIoT: Precision Agriculture





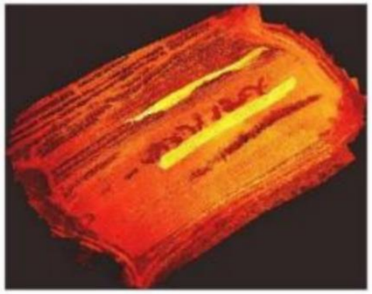
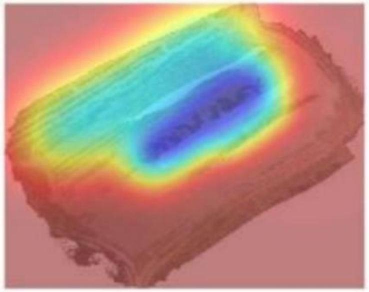
Scan farm using drone to capture low level details on daily basis, analyze differences each day and fuse with sensor information to identify areas that needs specific work

AgIoT: Vision

Ag Services

Yield estimation | Precision Irrigation | Pest Infection | Fertilizer application | …

Spatio-temporal view of the farm

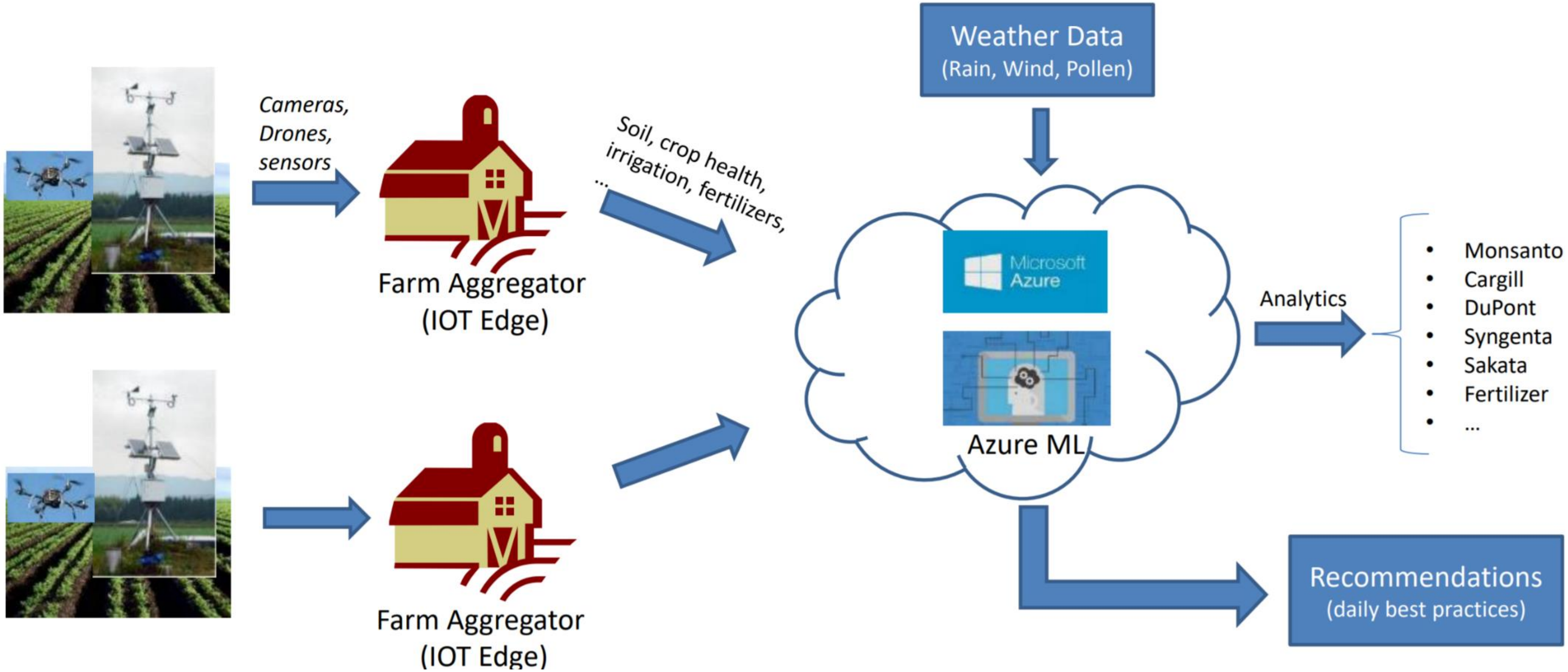




Sensors & UAVs



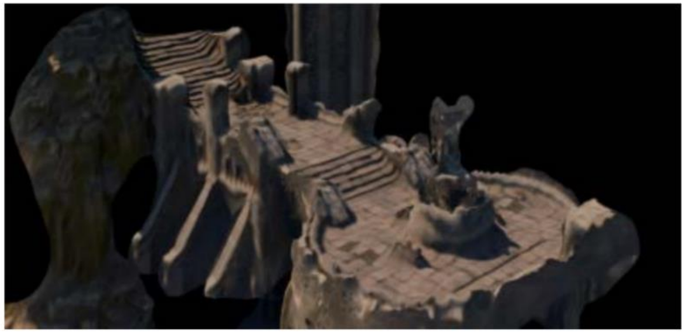


AgIoT: System Architecture

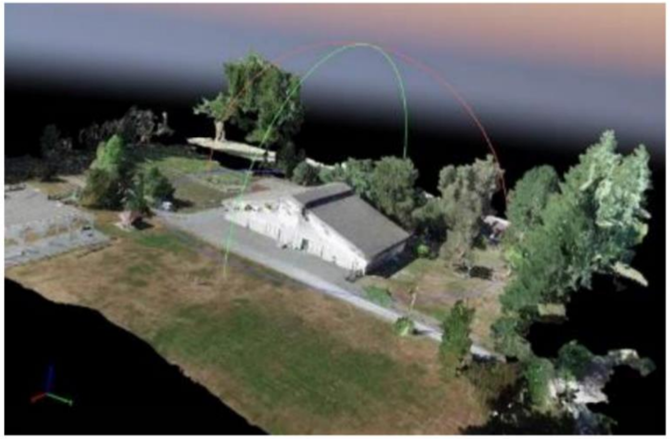


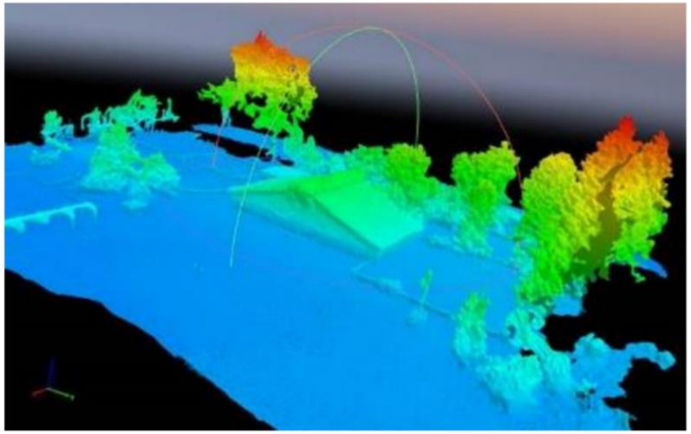
Autonomous 3D Scanning of Large Structures

3D reconstruction in simulator using a simulated drone flight



3D reconstruction in real world using actual drone





Expands AirSim AI Simulator to Include Autonomous Car Research

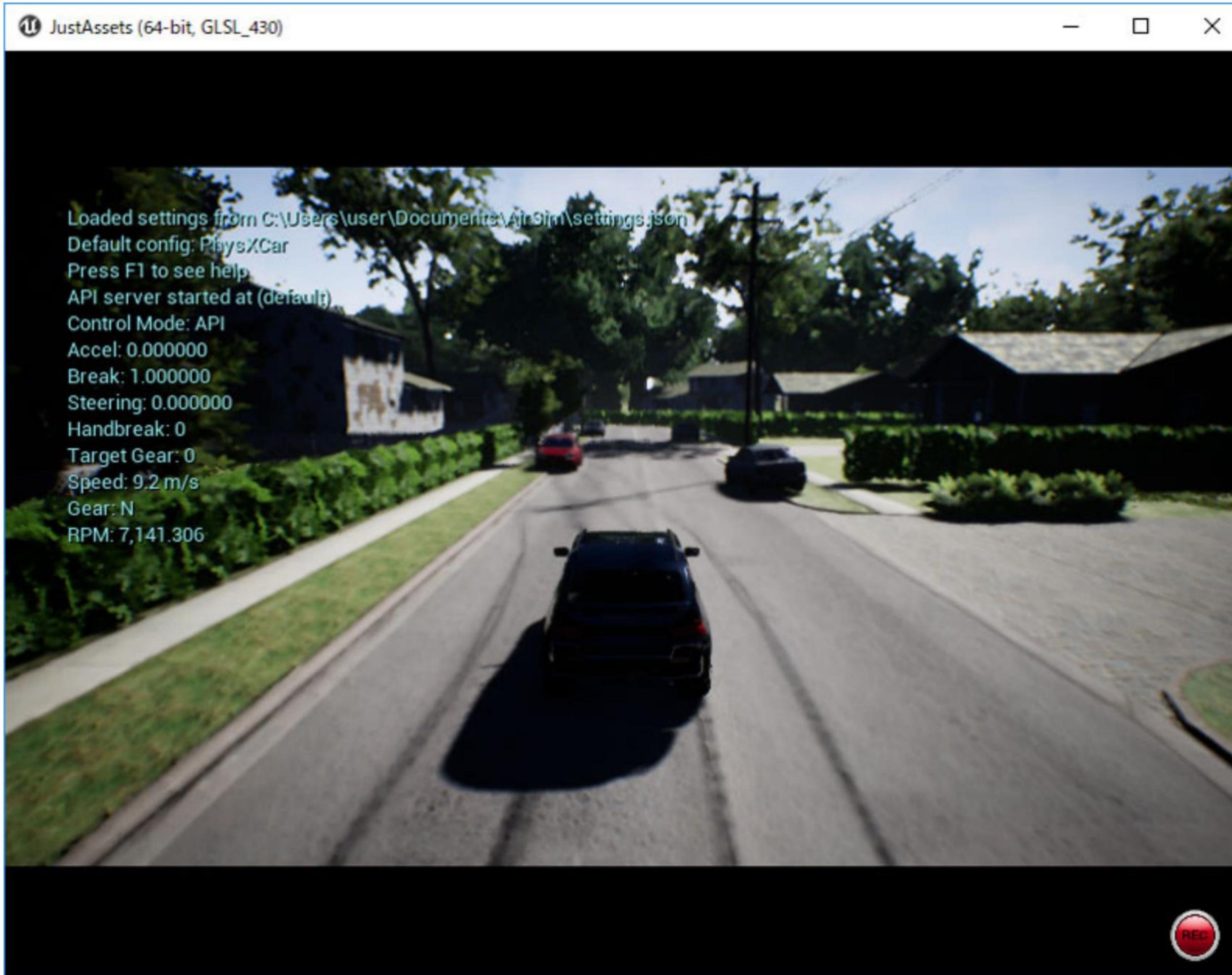
Penn Engineering

PRECISE

PENN RESEARCH IN EMBEDDED COMPUTING AND INTEGRATED SYSTEMS ENGINEERING

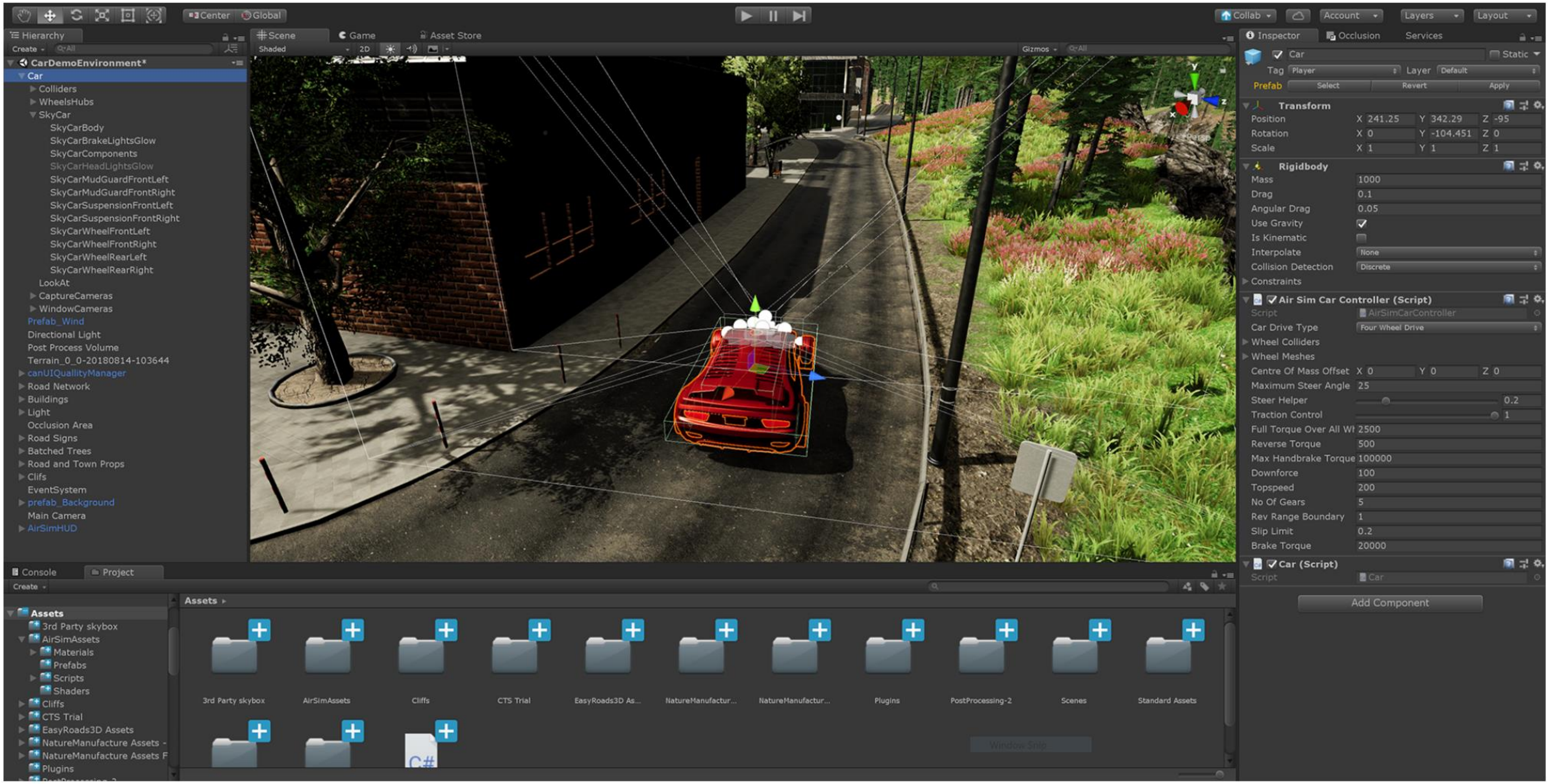
End-to-End Deep Learning for Autonomous Driving

Deep Reinforcement Learning for Autonomous Driving



Penn Engineering

AirSim on Unity



Penn Engineering

PRECISE

Thanks !