Anatomy of an Autonomous Vehicle

Instructor: Chris Mavrogiannis

TAs: Kay Ke, Gilwoo Lee, Matt Schmittle

*Slides based on or adapted from Sanjiban Choudhury

Today's objective

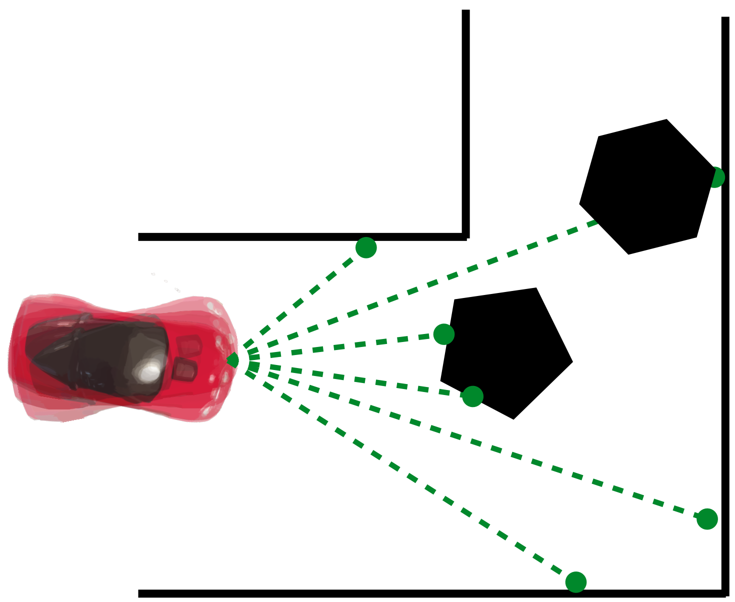
1. Learn how to architect a mobile robotic system

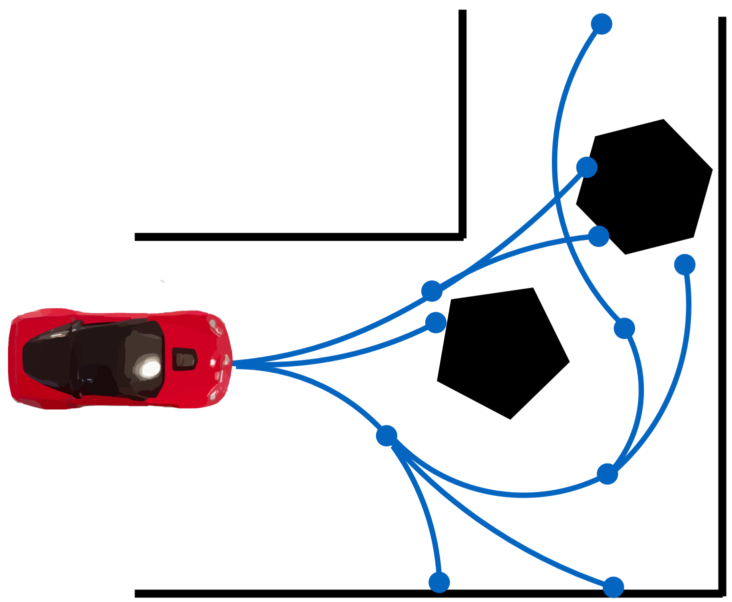
2. Step through a set of fundamental lessons that shape robot system / algorithm design

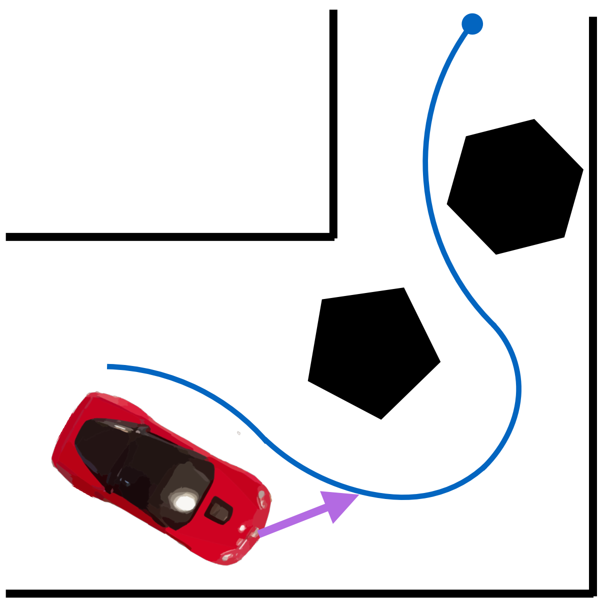
Estimate state

Plan a sequence of motions

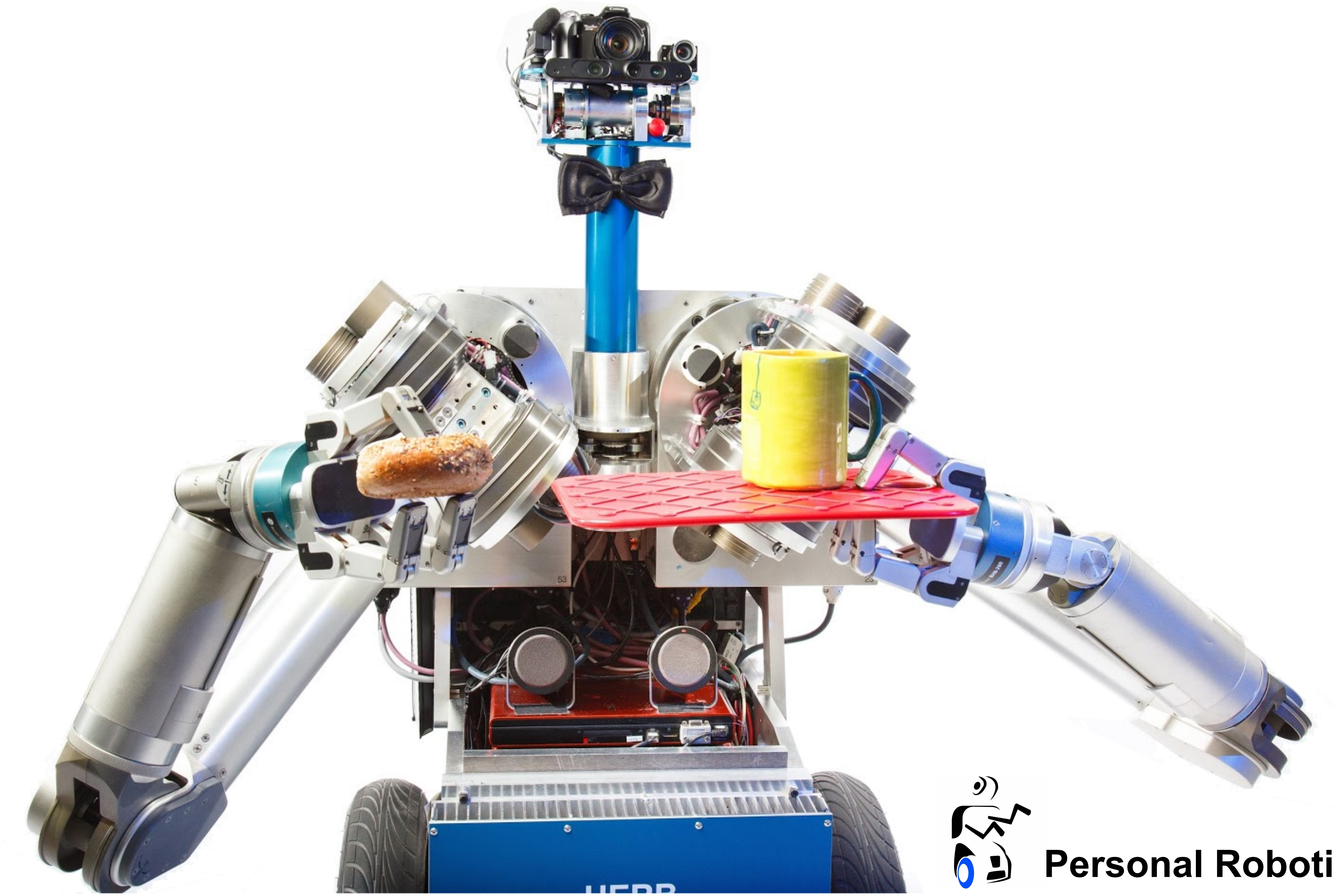
Control robot to follow plan





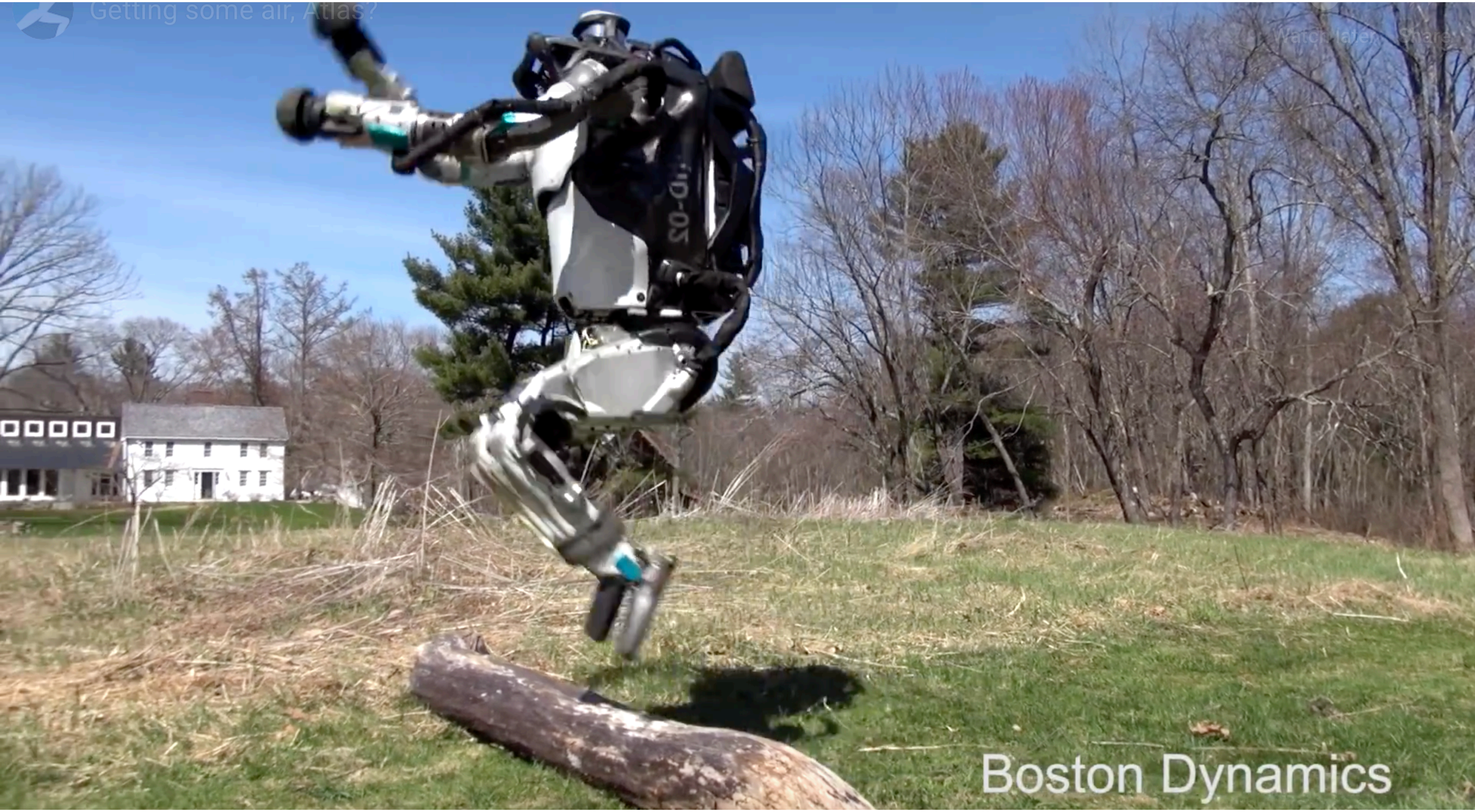


Mobile Manipulators



Personal Robotics Lab

Humanoids



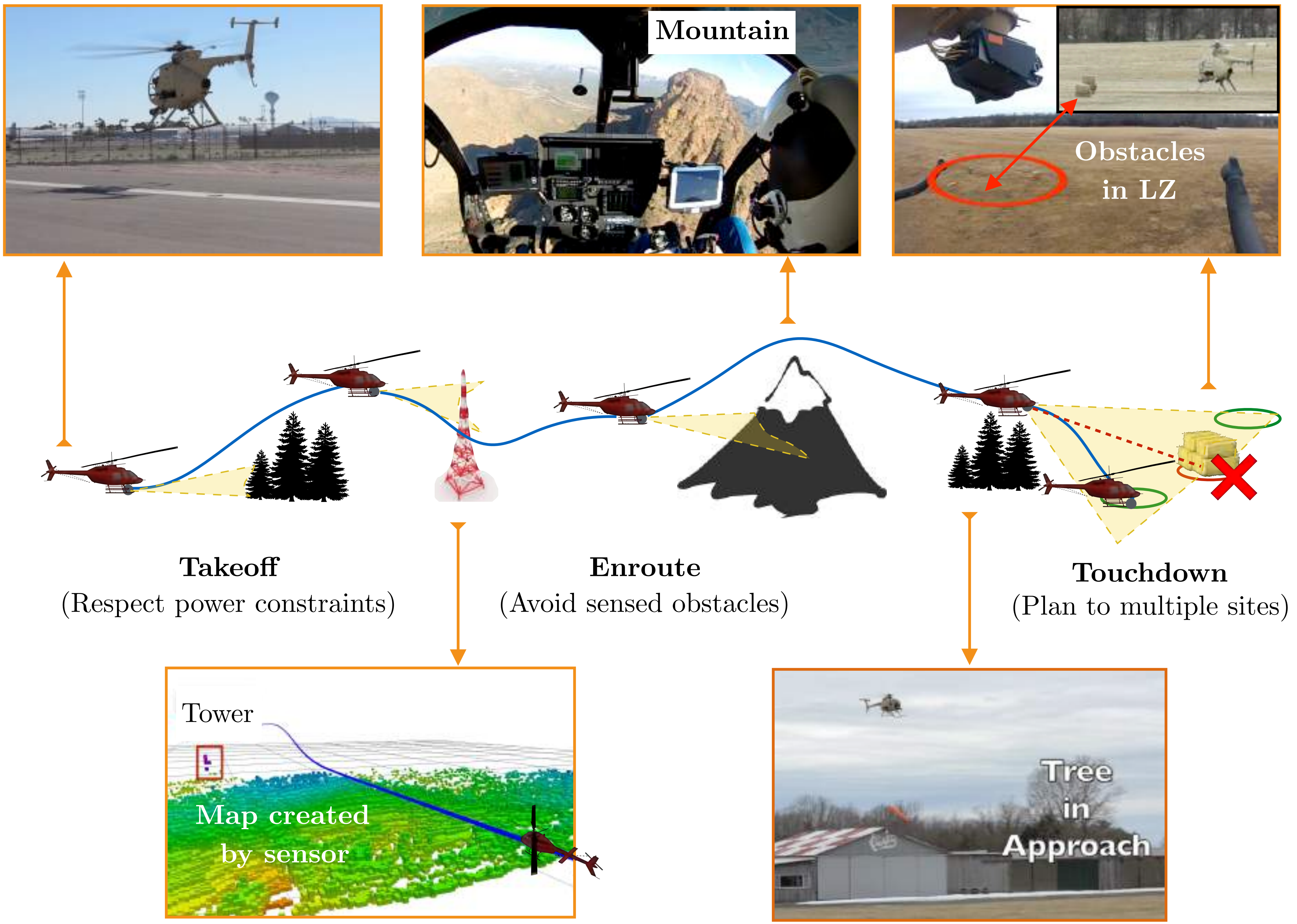
Self-driving Cars

Flying vehicles

Anatomy of a flying vehicle

Mission: Takeoff to Landing

Mission



Mountain
Obstacles in LZ
Takeoff
(Respect power constraints)
Enroute
(Avoid sensed obstacles)
Touchdown
(Plan to multiple sites)
Tower
Map created by sensor
Tree in Approach

Task: A contract the robot has to satisfy

Given:

- Start (latitude, longitude), Goal (latitude, longitude)
- List of no-fly-zones (unsafe air space)
- Coarse terrain map of continental USA
- Sensors - GPS, Laser, etc

Objective:

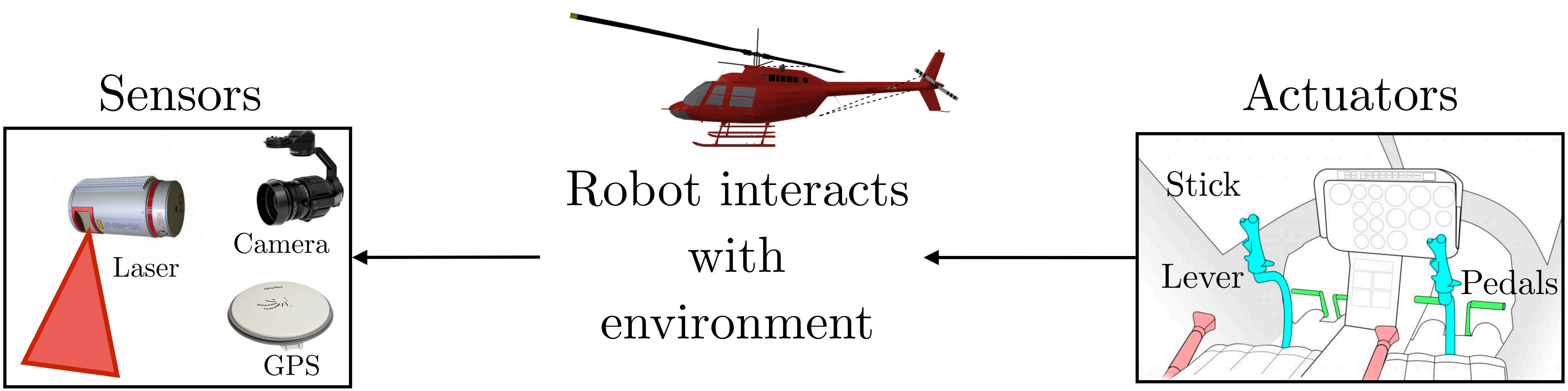
- Minimize time it takes to complete mission

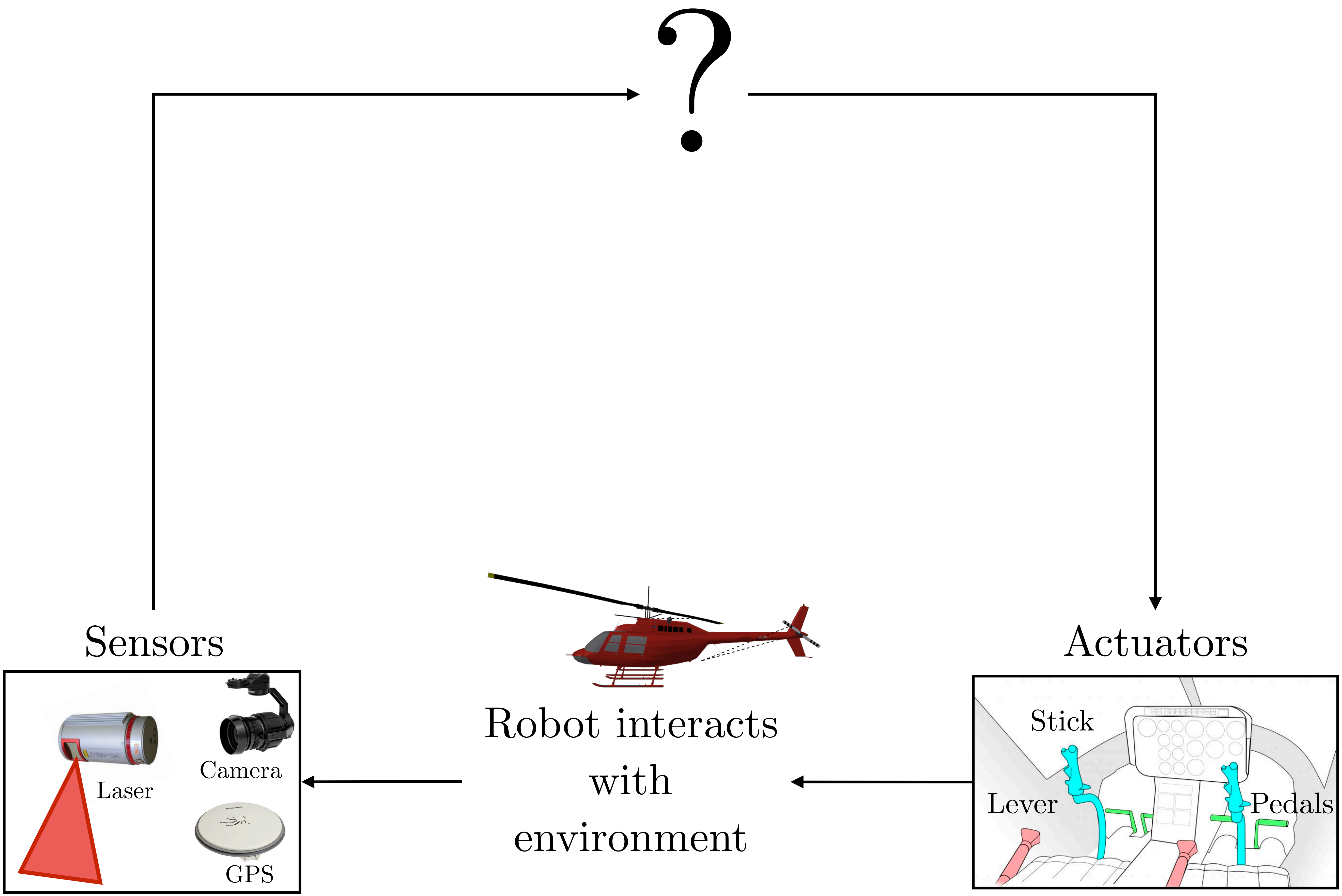
Constraint:

- Don't come close to obstacles / don't enter no-fly-zones
- Don't exceed limits of the vehicle (flying upside down)

How do we tractably solve the task?

Begin with a blank slate


Sensors
Laser
Camera
GPS
Robot interacts
with
environment
Actuators
Stick
Lever
Pedals


?
Sensors
Laser
Camera
GPS
Robot interacts
with
environment
Actuators
Stick
Lever
Pedals

Lesson 0: Look at one piece at a time

Q1: Assume we know everything about the world.

What commands should we send to the actuator?

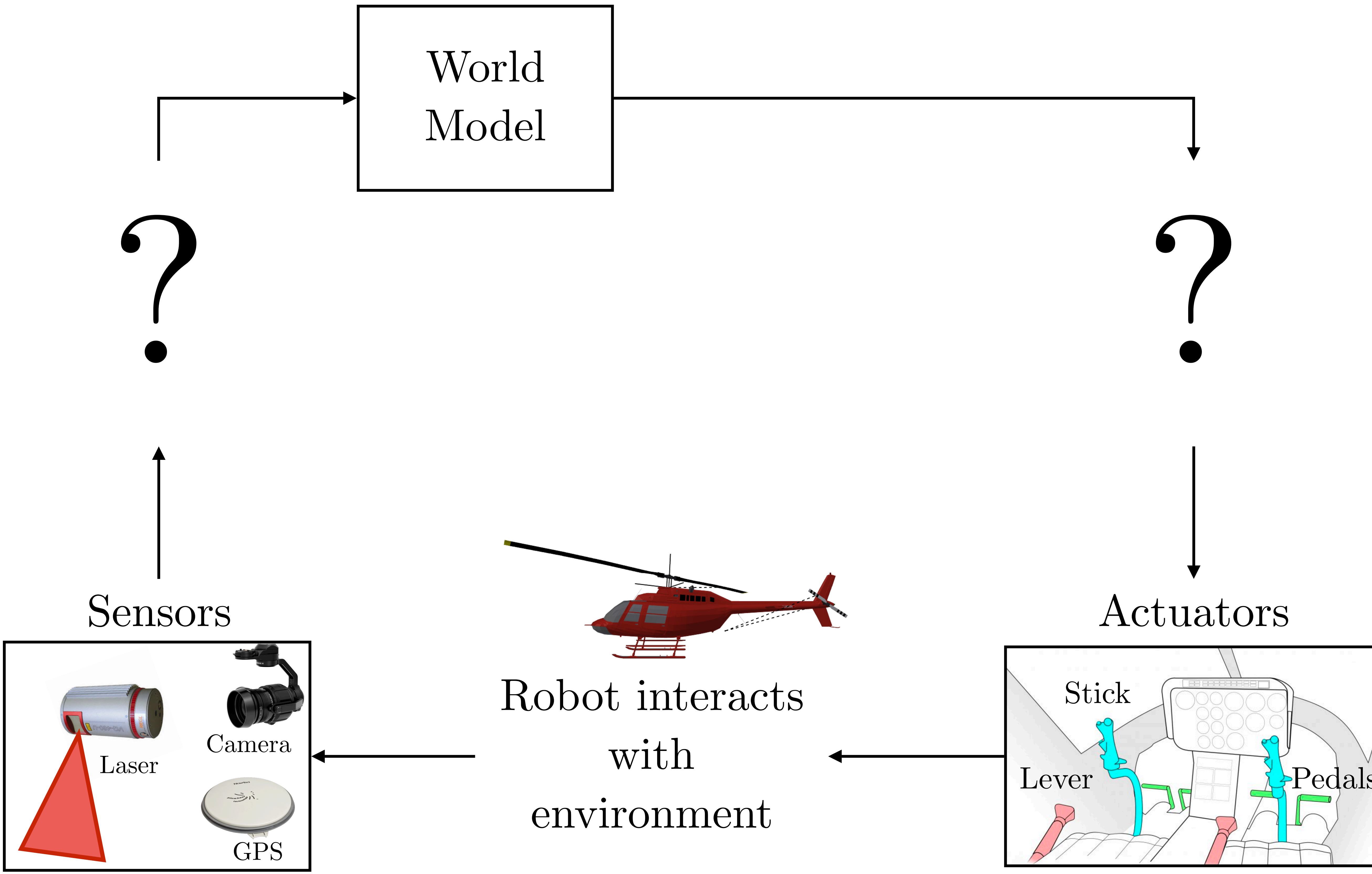
Q2: How do we use raw sensor data to update what we know about the world?

Look at one piece at a time

Q1: Assume we know everything about the world.

What commands should we send to the actuator?

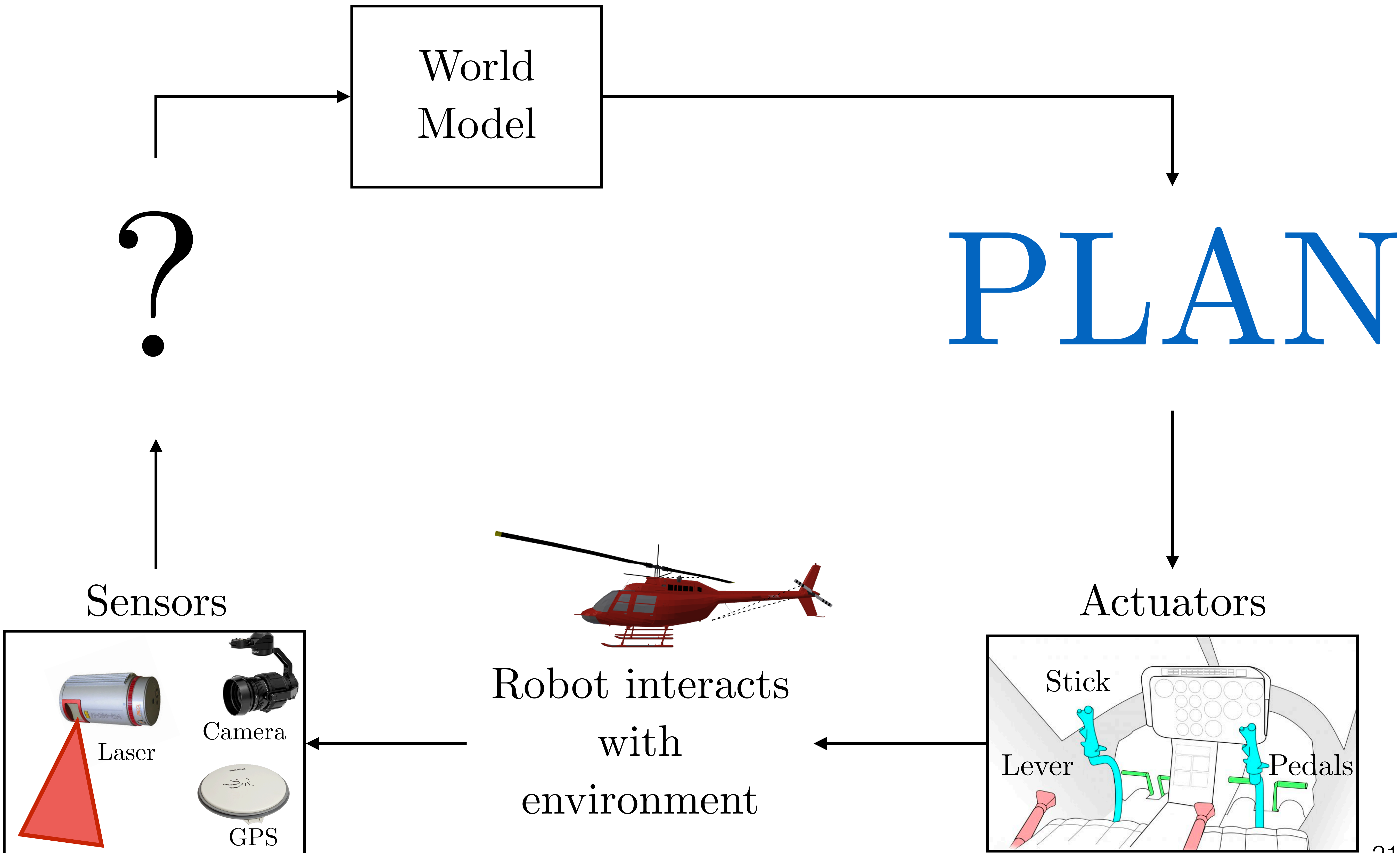
Q2: How do we use raw sensor data to update what we know about the world?


World
Model
?
?
Sensors
Laser
Camera
GPS
Robot interacts
with
environment
Actuators
Stick
Lever
Pedals

What is the world model?

List of everything we need to know to accomplish the task

- Where is the robot in the world? What is its state?
- What are the obstacles in the world?
- What type are the obstacles (radio towers, trees)?
- What are the no-fly-zones?
- Are there other aircrafts?
- What is the wind, temperature, etc?


World
Model
?
PLAN
Sensors
Laser
Camera
GPS
Robot interacts
with
environment
Actuators
Stick
Lever
Pedals

What is planning?

Planning is an optimization problem in which ...

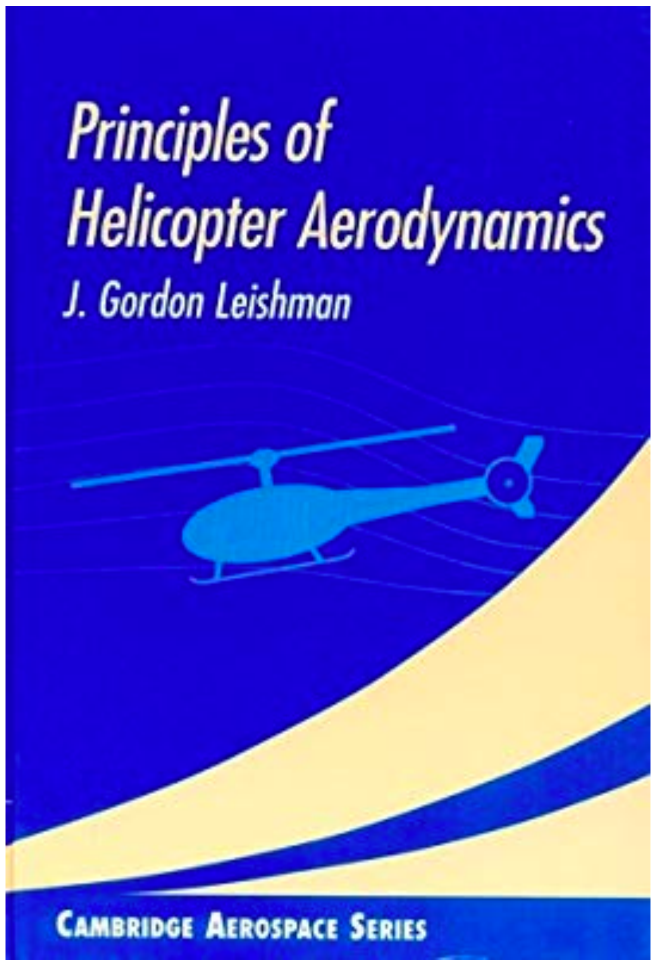
we search over a sequence of actions...

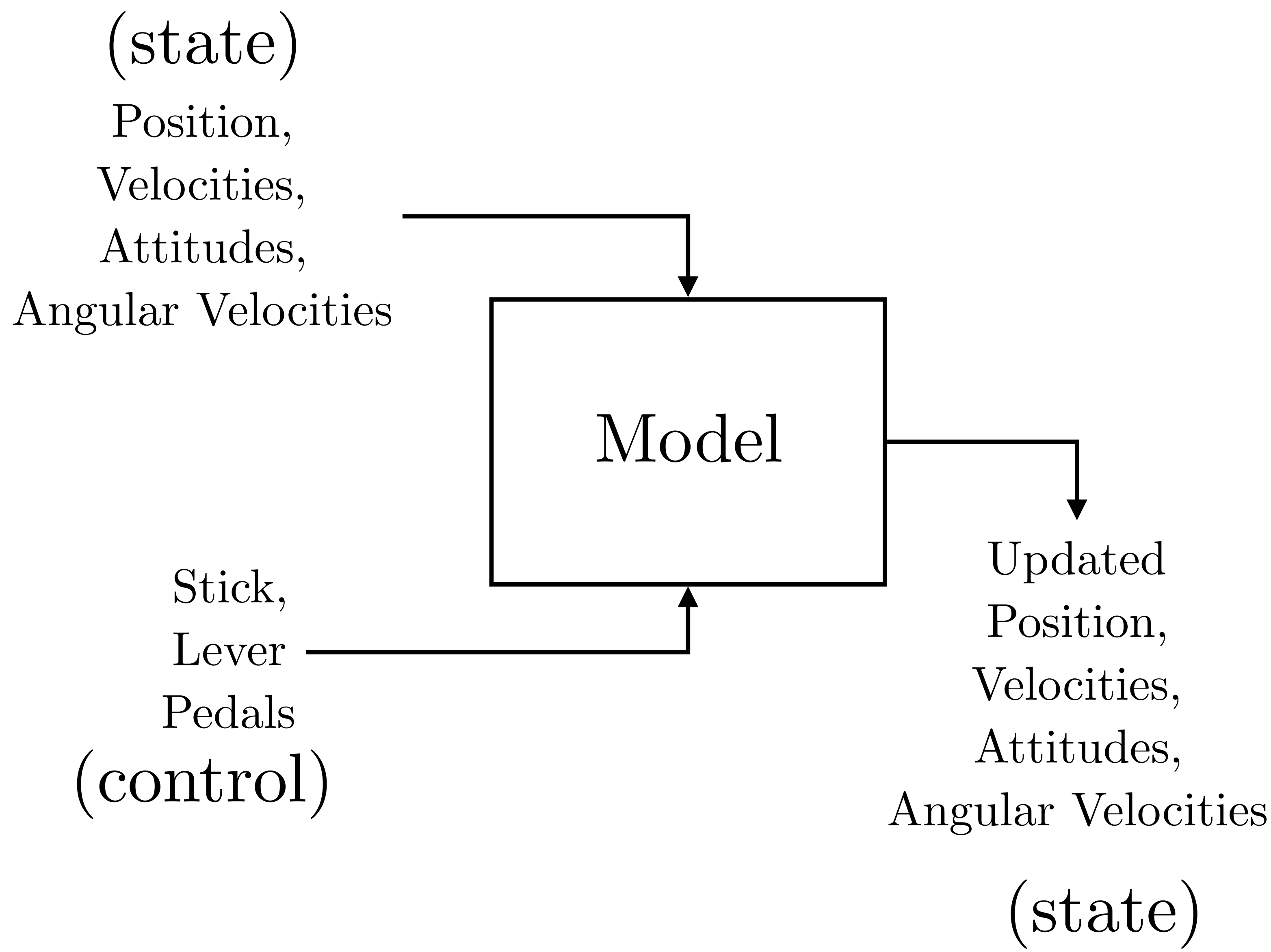
towards minimizing a cost function (e.g., time)...

using a model of the robot to predict where it will go...

while making sure we are not violating constraints (e.g. crash).

How do we get a model of the helicopter?





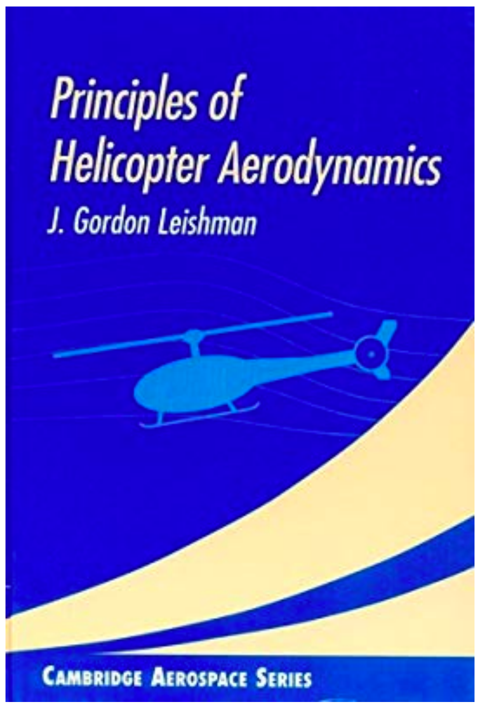
Problem: Model is very complicated! Intractable to plan with it.

Insight

"All models are **wrong**, but some are **useful**"

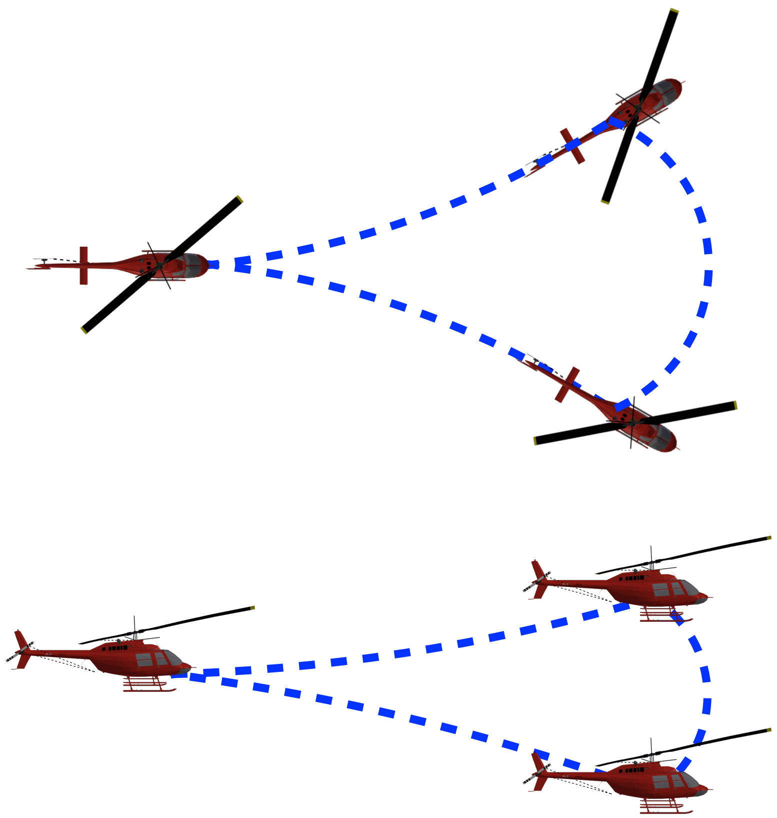
-George Box

Lesson 1: Plan with simple models



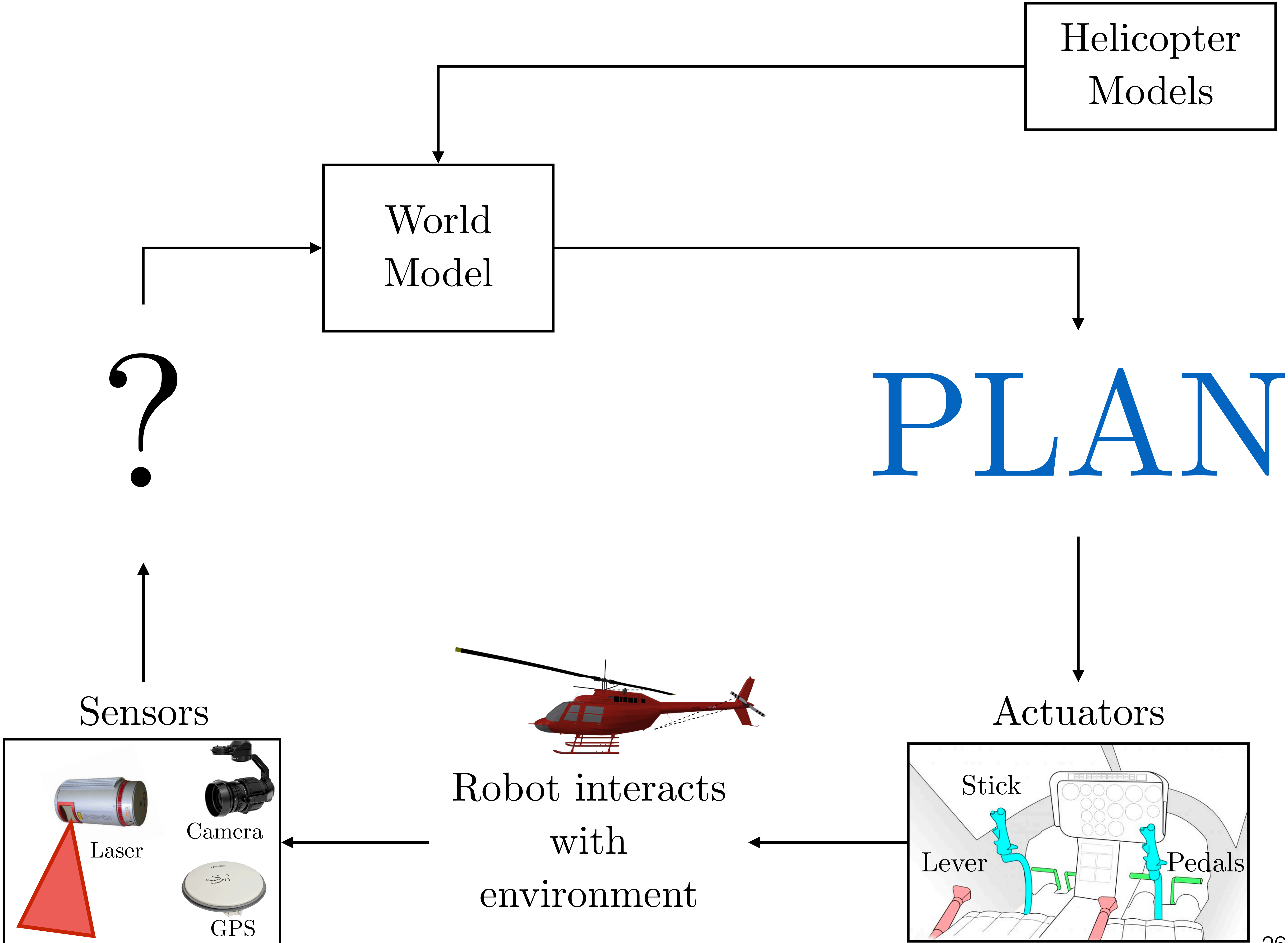
Complex aerodynamical model

Use domain knowledge to simplify model

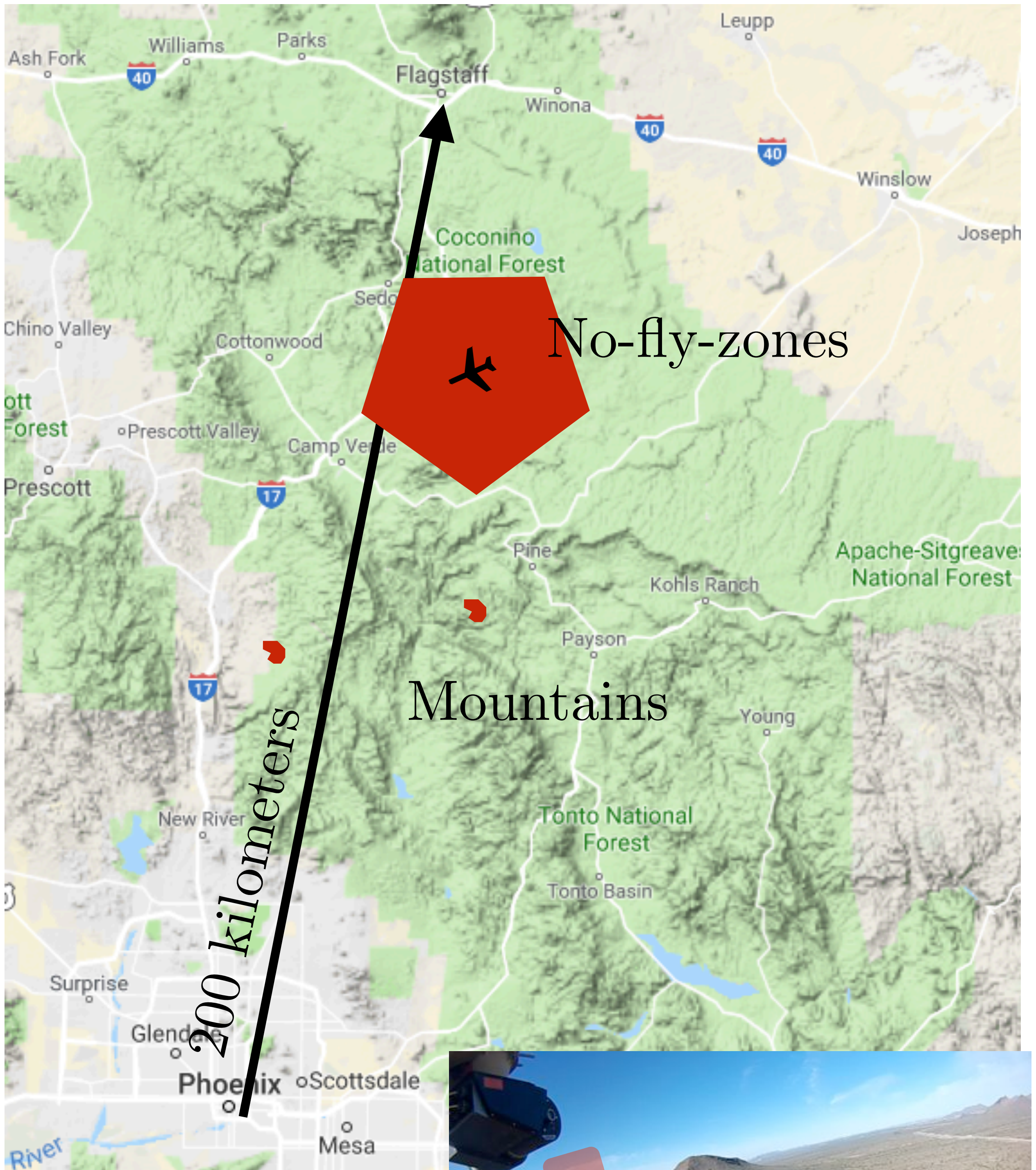


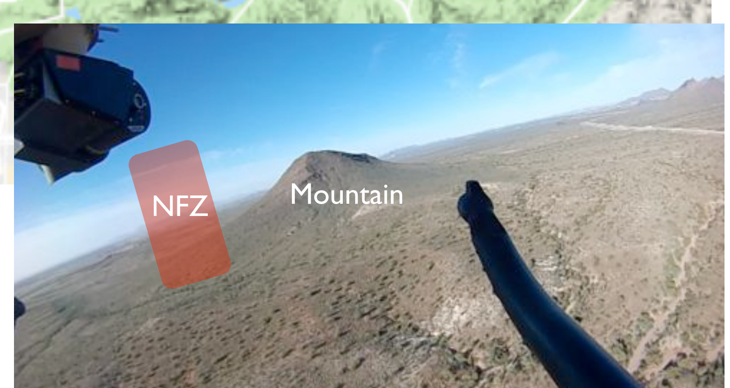
Flying unicycle at high speeds!

Different models at different flight regimes


Helicopter Models
World Model
?
PLAN
Sensors
Laser
Camera
GPS
Robot interacts with environment
Actuators
Stick
Lever
Pedals

What resolution should we plan at?





Example mission:

Fly from Phoenix to Flagstaff as fast as possible (200 km)

Avoid mountains, no-fly-zones, radio towers, wires, bad weather

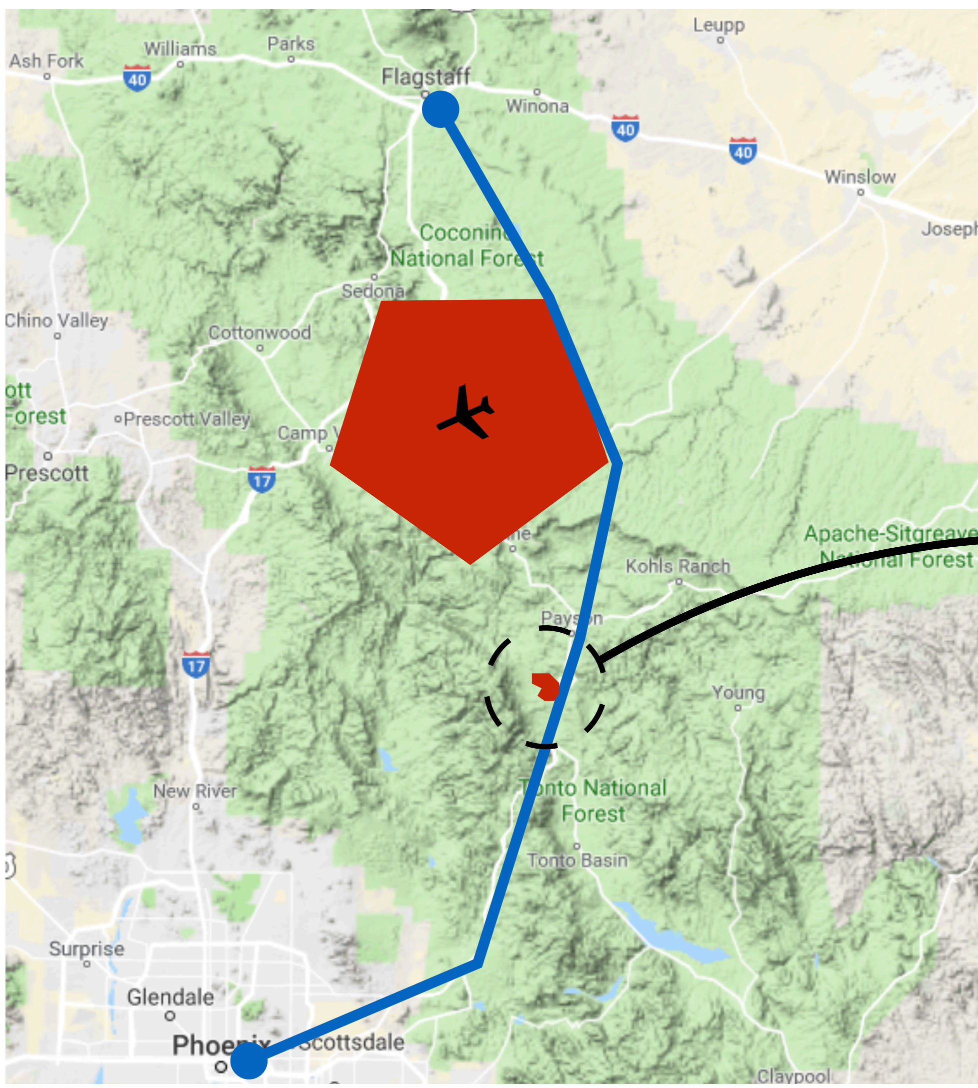
Pass through narrow gaps

Problem:

Take forever to plan at high resolution ALL the way to goal

Lesson 2: Plan at multiple resolutions

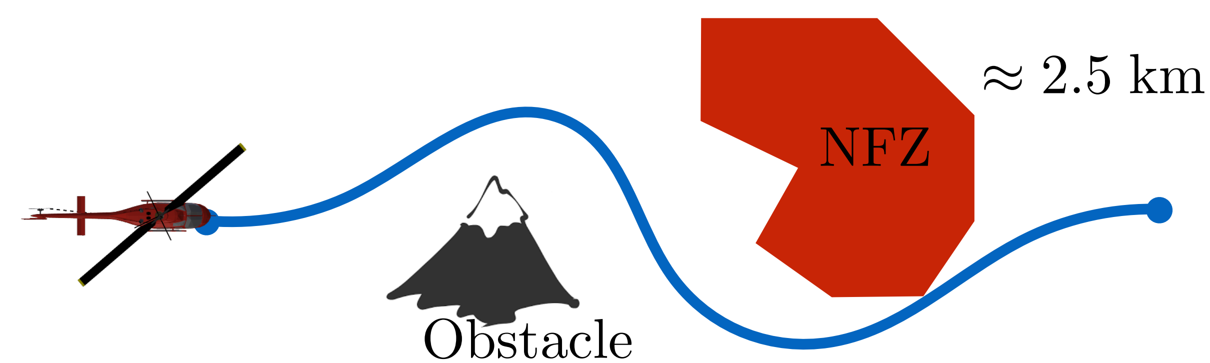
Global planner



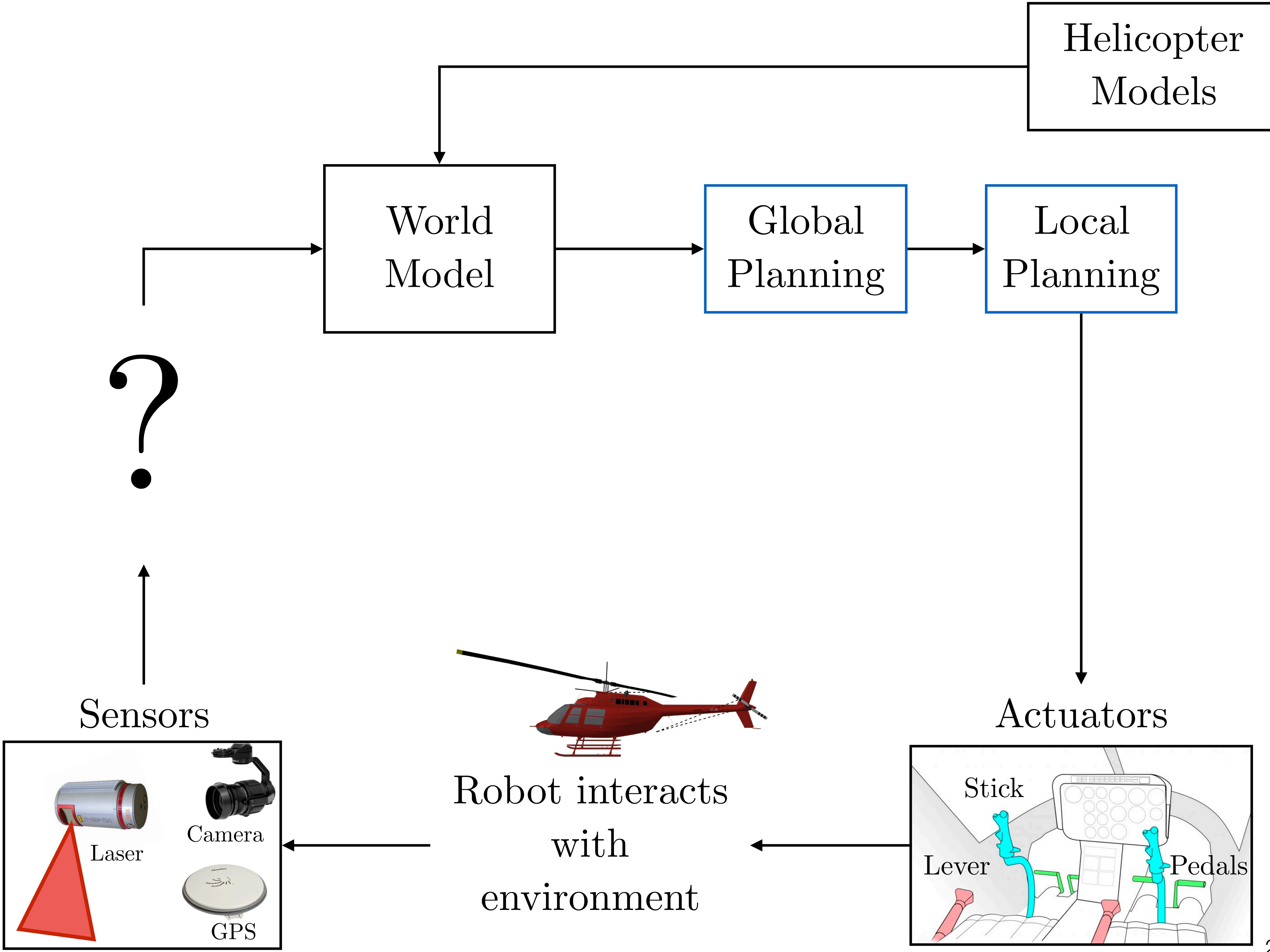
Plan at coarse (1km) resolution,
compute entire route from start to goal
avoid large obstacles, no-fly-zones etc

(only consider factors that significantly affect mission time)

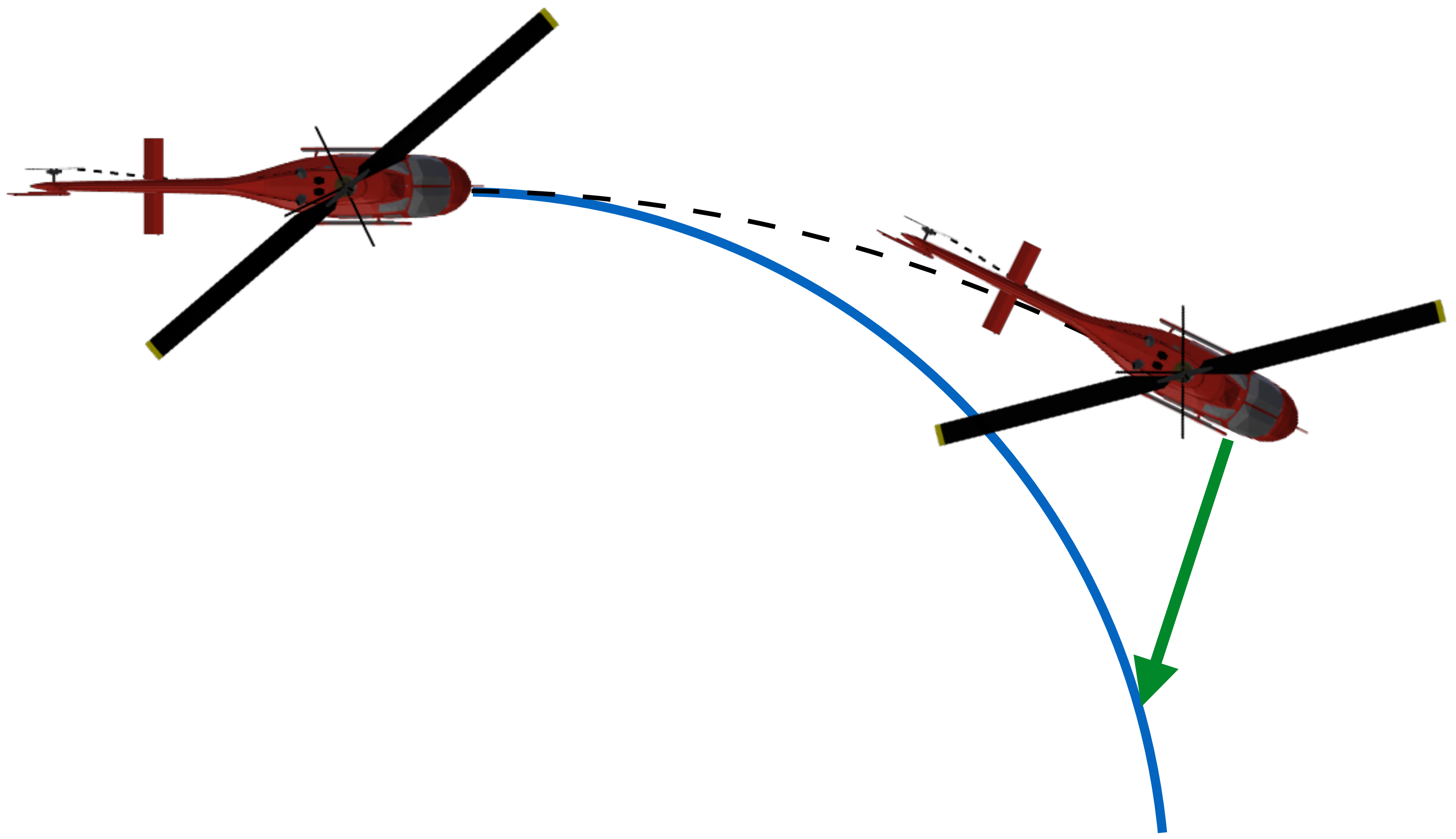
Local planner



Plan at high (10 m) resolution,
follow the global route,
avoid all obstacles, produce smooth dynamically feasible paths

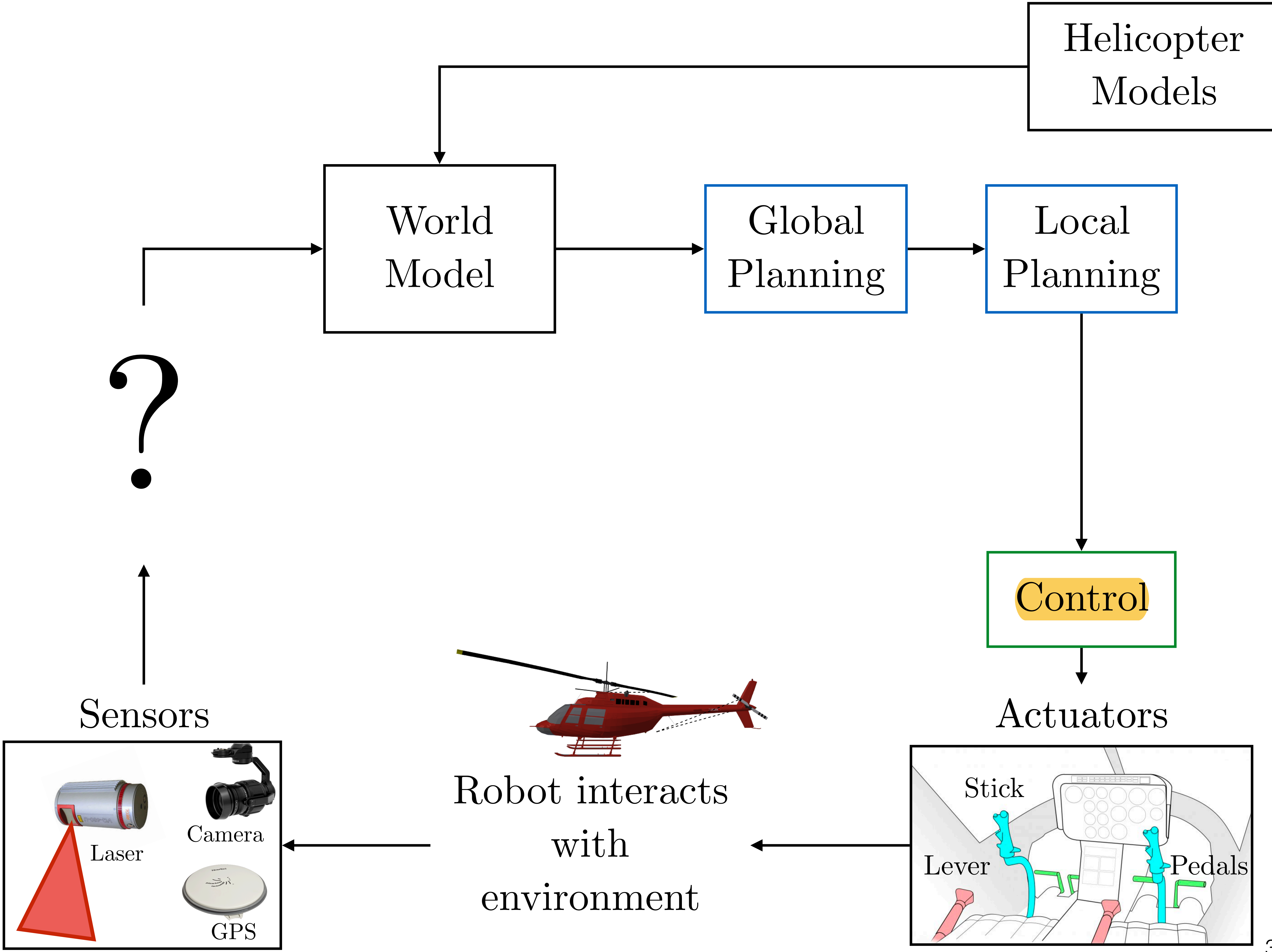

Helicopter
Models
World
Model
Global
Planning
Local
Planning
?
Sensors
Laser
Camera
GPS
Robot interacts
with
environment
Actuators
Stick
Lever
Pedals

Lesson 3: Open loop planning is not enough



Robot will go "off" the plan for many reasons
(disturbance, model errors, actuation errors, ...)

A **controller** immediately corrects for any tracking error
and gets the robot back on the path


Helicopter Models
World Model
Global Planning
Local Planning
?
Control
Sensors
Actuators
Robot interacts with environment
Laser
Camera
GPS
Stick
Lever
Pedals

Look at one piece at a time

Q1: Assume we know everything about the world.

What commands should we send to the actuator?

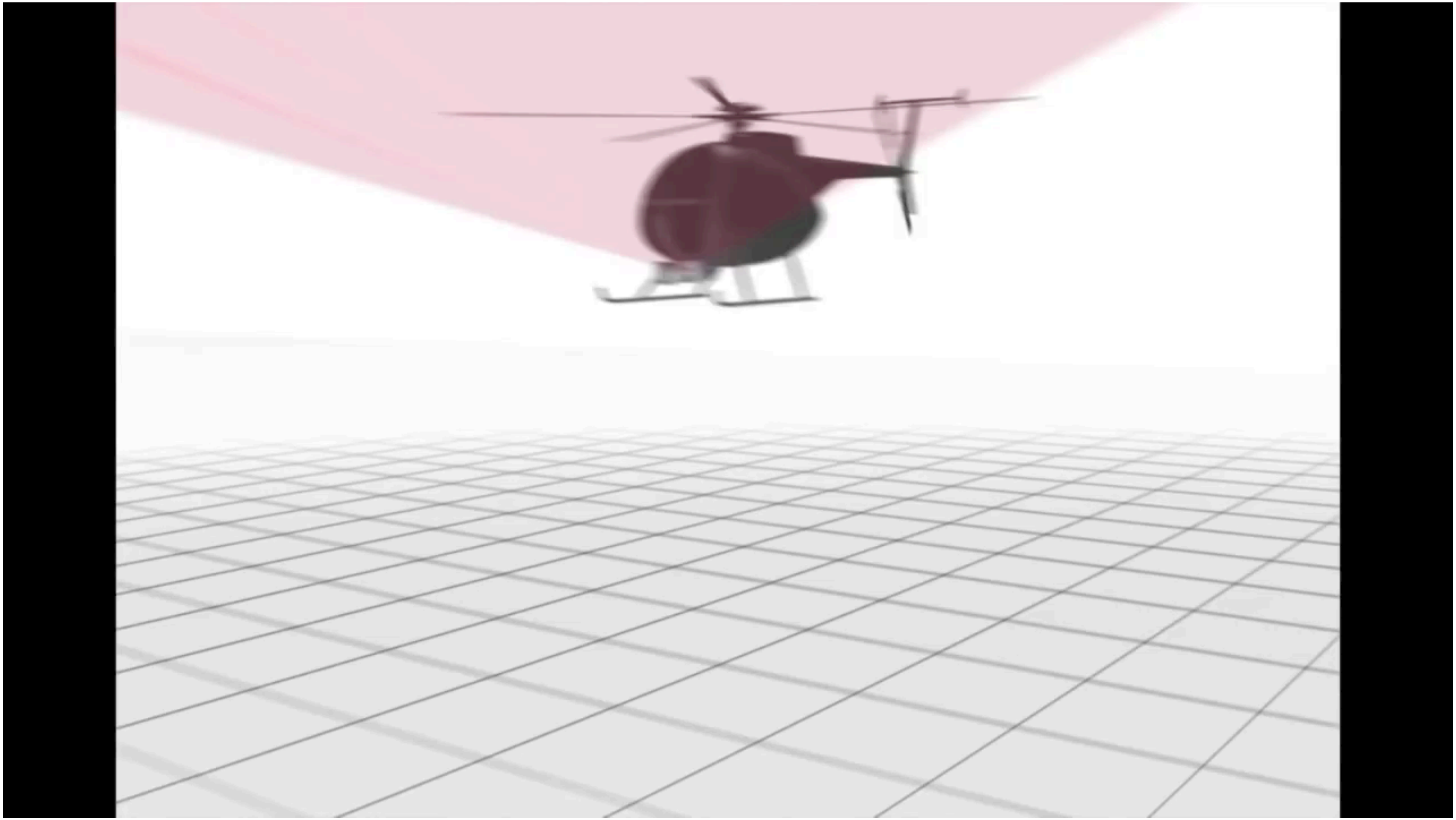
$\boxed{\text{Q2: How do we use raw sensor data to update what we know about the world?}}$

What is the world model?

List of everything we need to know to accomplish the task

- Where is the robot in the world? What is it's state?	GPS
- What are the obstacles in the world?	Laser
- What type are the obstacles (radio towers, trees)?	Camera
- What are the no-fly-zones?	Radio
- Are there other aircrafts?	Radio
- What is the wind, temperature, etc?	Pitot tube, barometer, etc

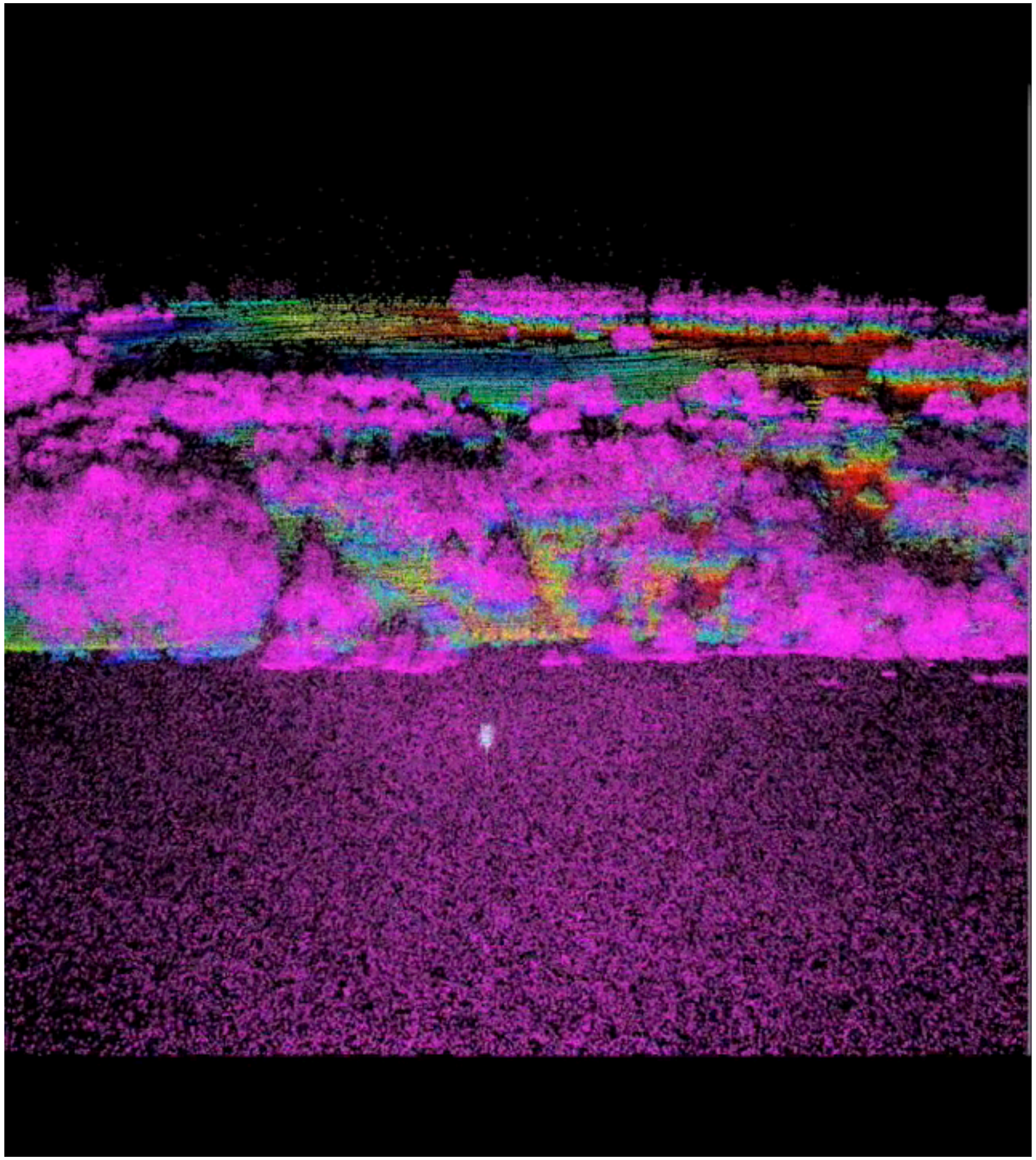
Can we simply "fuse" laser readings to map the world?



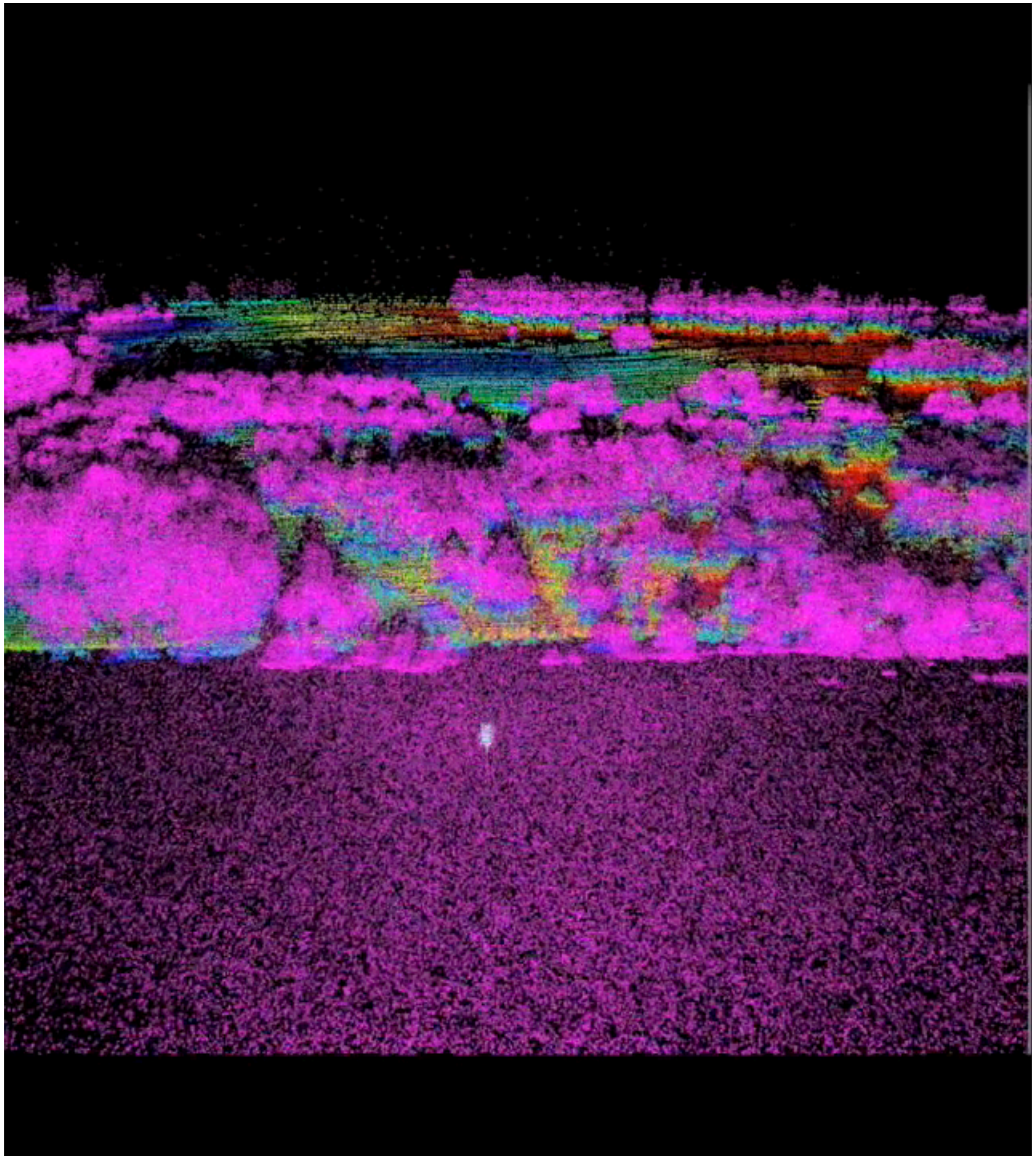
(courtesy Chamberlain et al.)

Flying in a snow storm

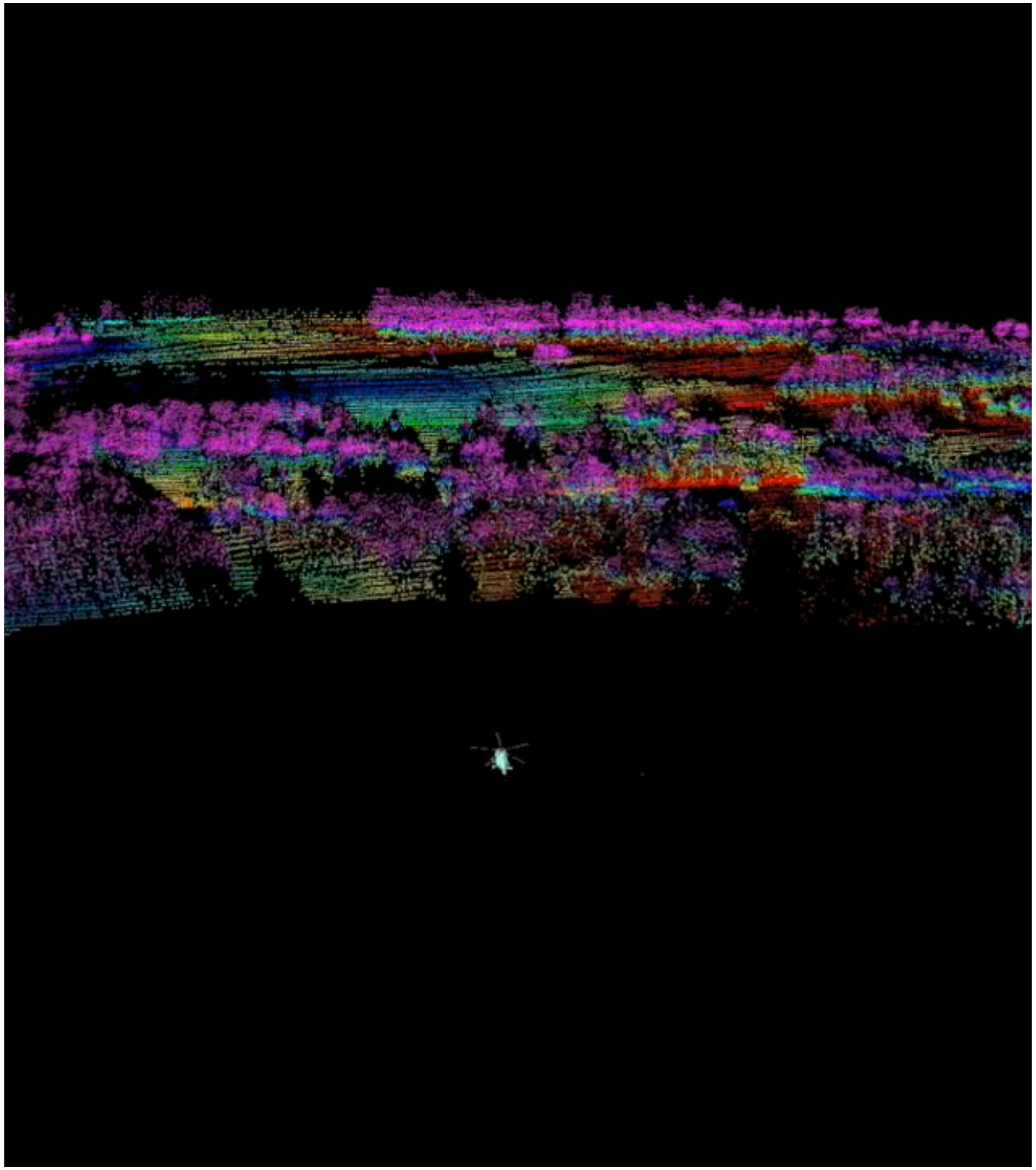
Flying in a snow storm



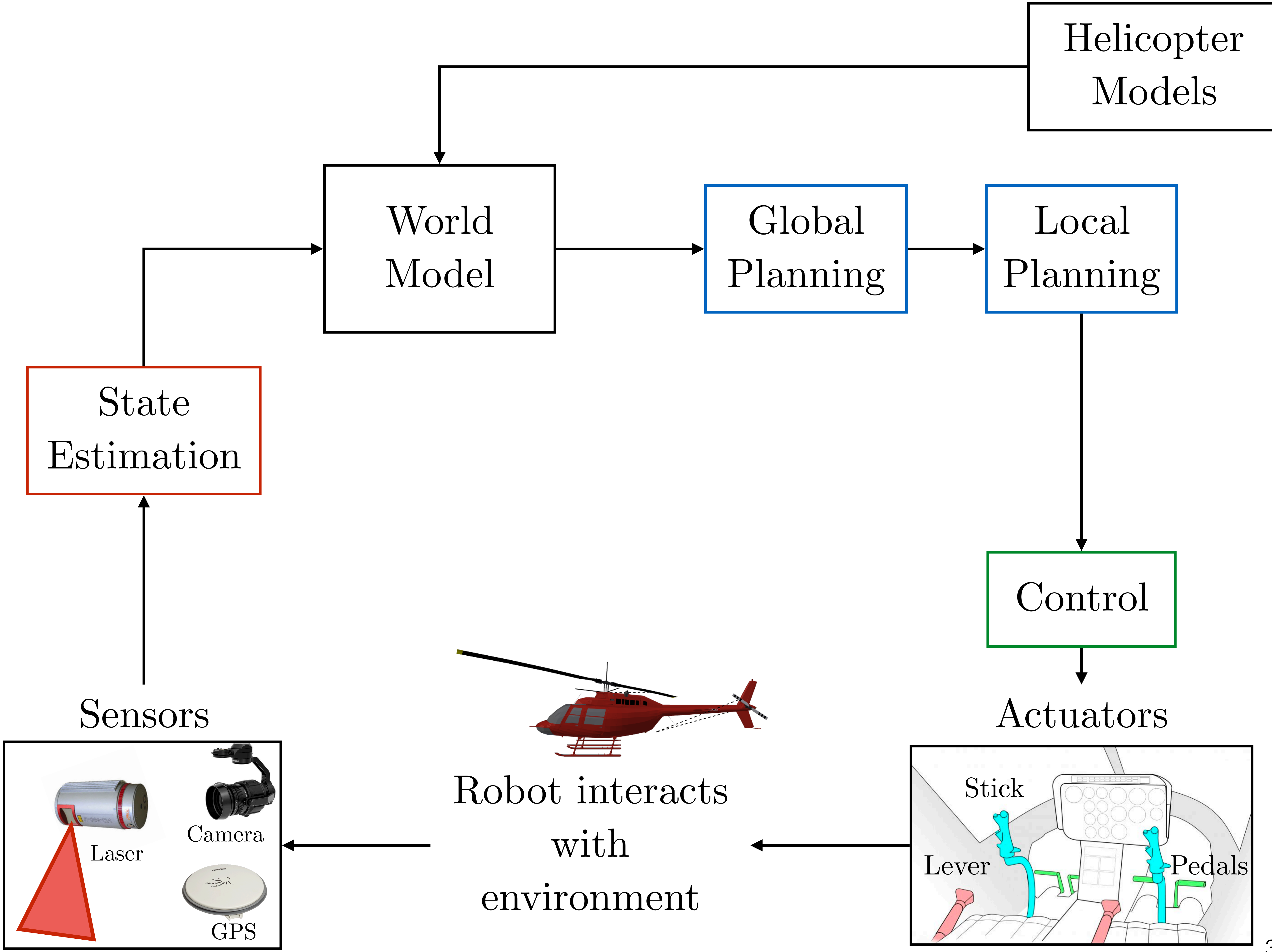
Lesson 4: Use probabilistic models of the sensor



Laser reflected by snow!



Correctly fused laser data using probabilistic models

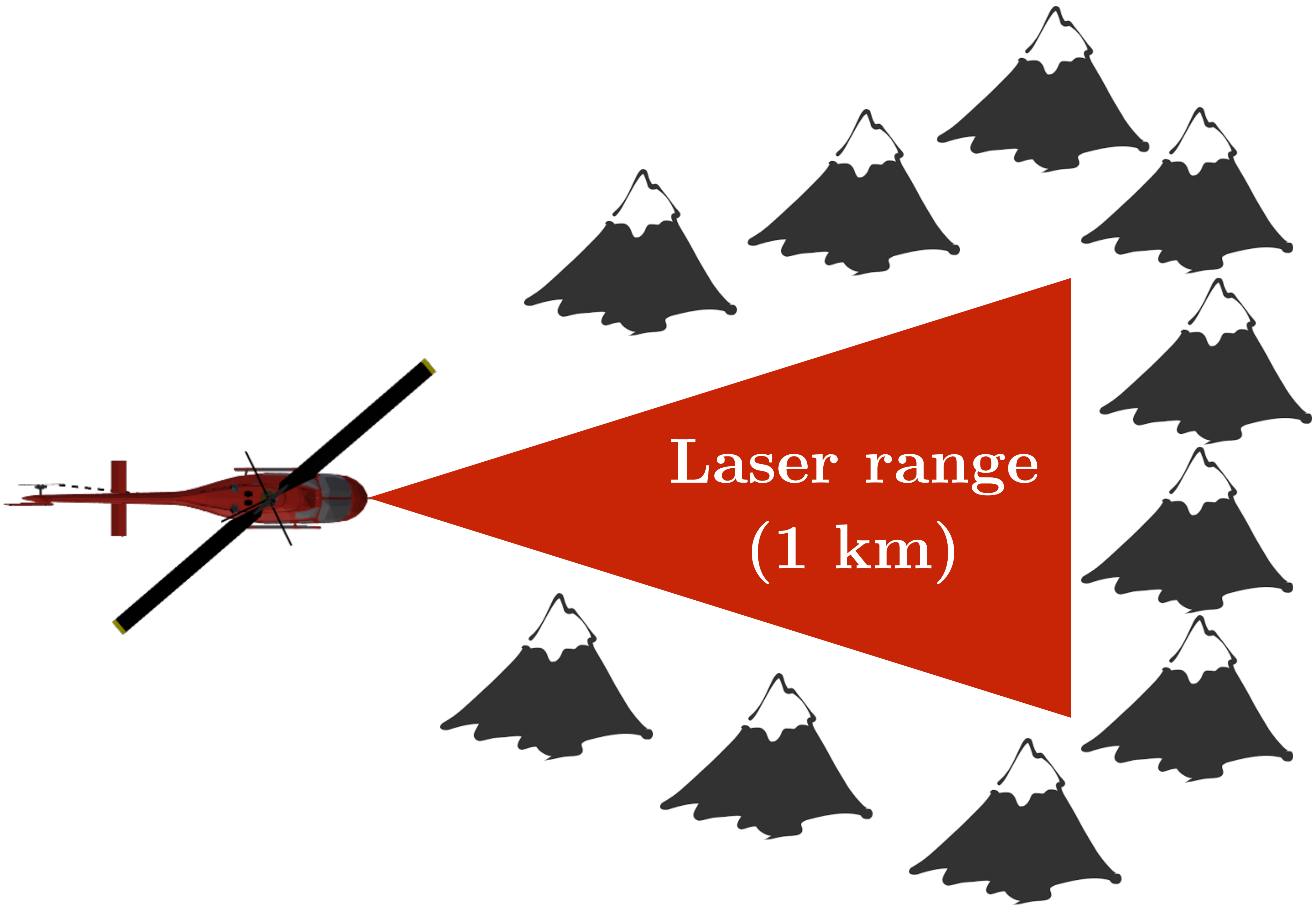

Helicopter Models
World Model
Global Planning
Local Planning
State Estimation
Control
Sensors
Actuators
Laser
Camera
GPS
Robot interacts with environment
Stick
Lever
Pedals

What is state estimation?

Given raw sensor data, use probabilistic models to estimate world model

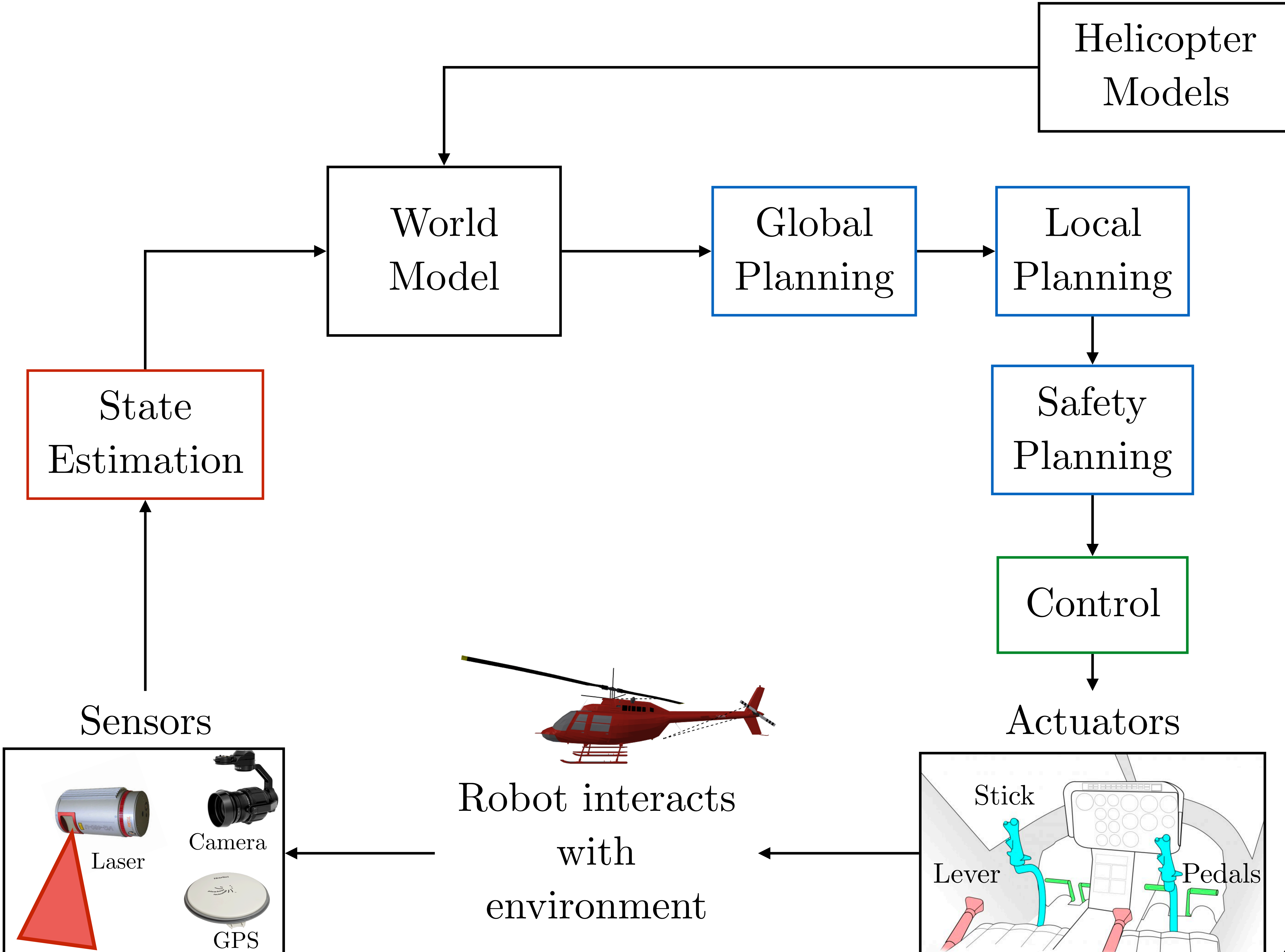
$$P(\text{world model}|\text{data})$$

Lesson 5: Guarantee safety



What if the robot encounters unexpected obstacles?

Safety planner that guarantees the robot can stay safe


Helicopter Models
World Model
Global Planning
Local Planning
State Estimation
Safety Planning
Control
Sensors
Actuators
Laser
Camera
GPS
Robot interacts with environment
Stick
Lever
Pedals

Today's objective

1. Learn how to architect a mobile robotic system

2. Step through a set of fundamental lessons that shape robot system / algorithm design

Anatomy of a self-driving car

BOSS: CMU's winning entry to DARPA challenge

(2007)

Tartan Racing: A Multi-Modal Approach to the DARPA Urban Challenge

April 13, 2007

Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, John Dolan, Dave Duggins, Dave Ferguson, Tugrul Galatali, Chris Geyer, Michele Gittleman, Sam Harbaugh, Martial Hebert, Tom Howard, Alonzo Kelly, David Kohanbash, Maxim Likhachev, Nick Miller, Kevin Peterson, Raj Rajkumar, Paul Rybski, Bryan Salesky, Sebastian Scherer, Young Woo-Seo, Reid Simmons, Sanjiv Singh, Jarrod Snider, Anthony Stentz, William "Red" Whittaker, and Jason Ziglar

Carnegie Mellon University

Hong Bae, Bakhtiar Litkouhi, Jim Nickolaou, Varsha Sadekar, and Shuqing Zeng

General Motors

Joshua Struble and Michael Taylor

Caterpillar

Michael Darms

Continental AG

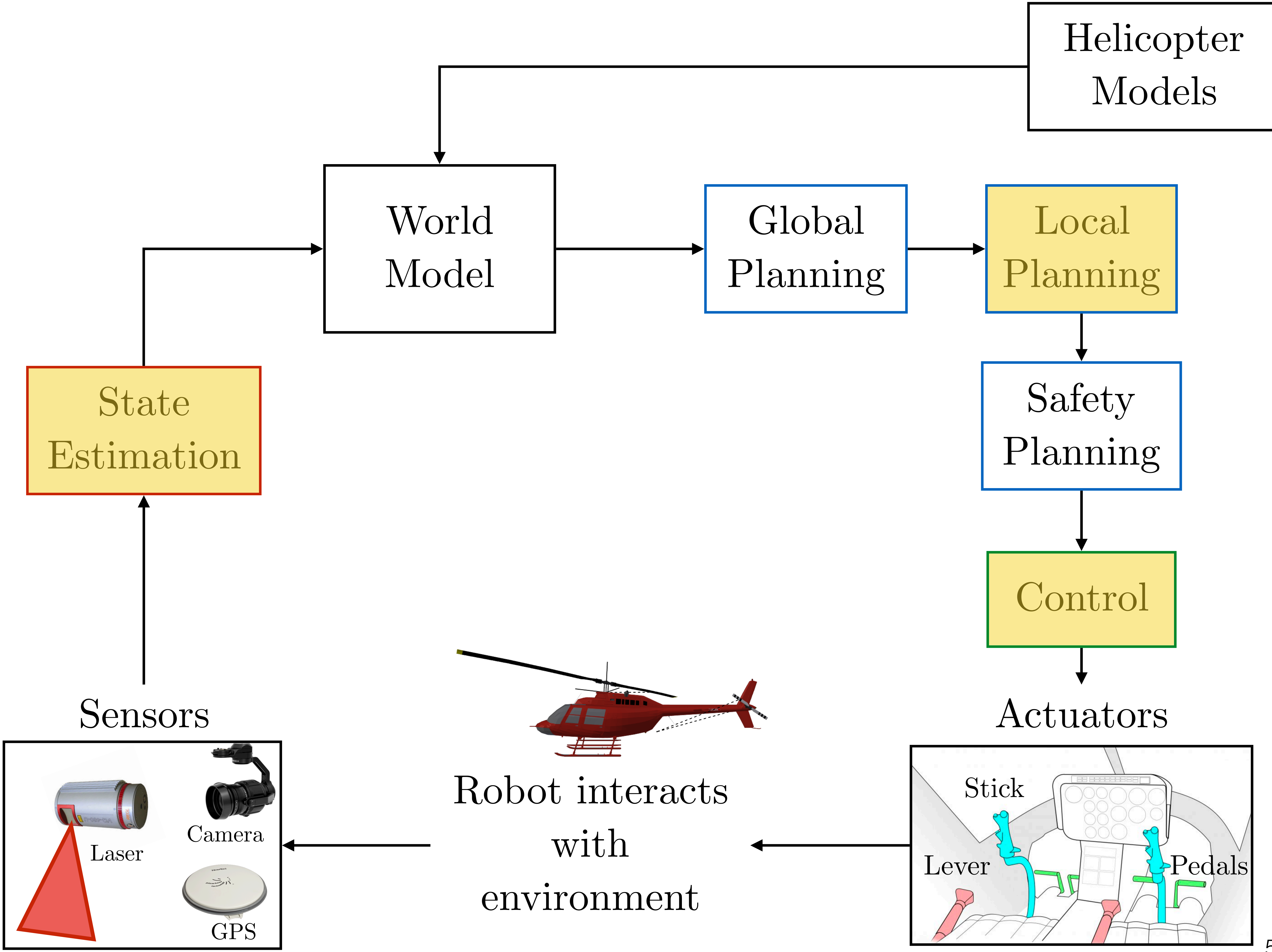
http://repository.cmu.edu/cgi/viewcontent.cgi?article=1967&context=robotics

BOSS in action!

1. World Model
2. Car Model
3. State Estimation
4. Global Planner
5. Local Planner
6. Safety Planner
7. Control

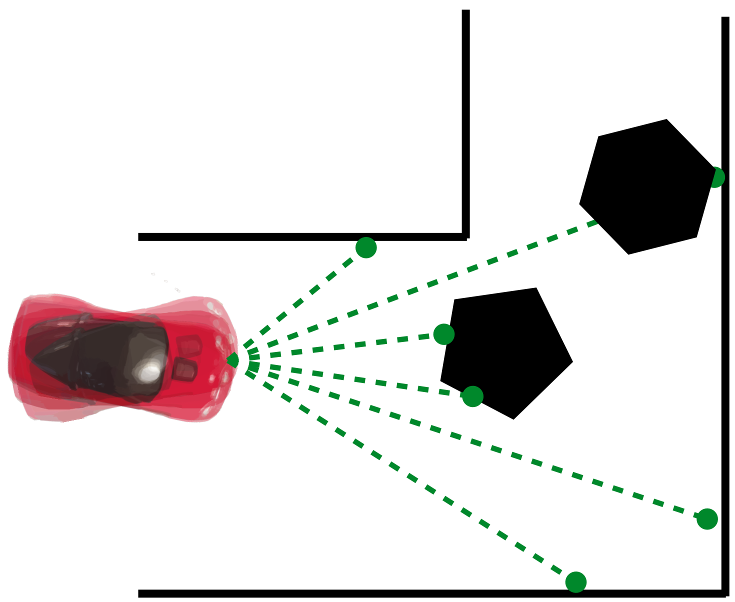
Additional challenges: Predict human drivers

Looking ahead...


Helicopter Models
World Model
Global Planning
Local Planning
State Estimation
Safety Planning
Control
Sensors
Laser
Camera
GPS
Robot interacts with environment
Actuators
Stick
Lever
Pedals

Next lecture: Introduction to State Estimation

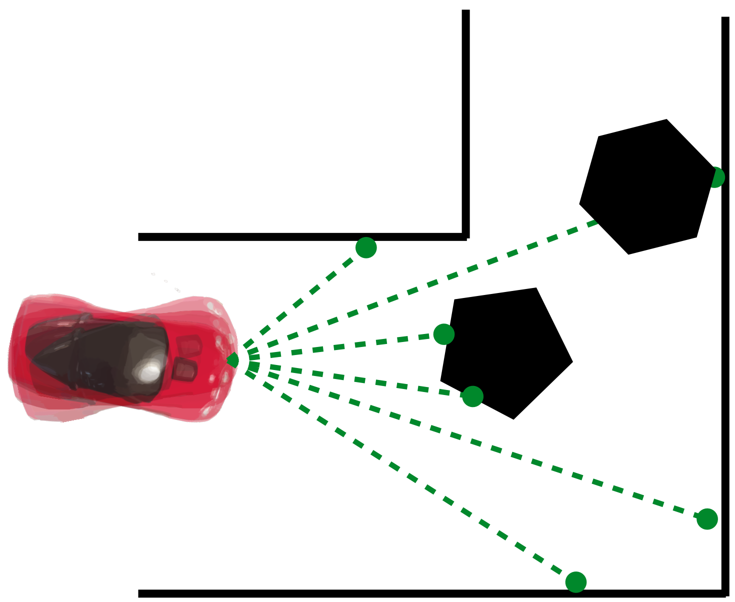
Estimate state



Weeks 2-4

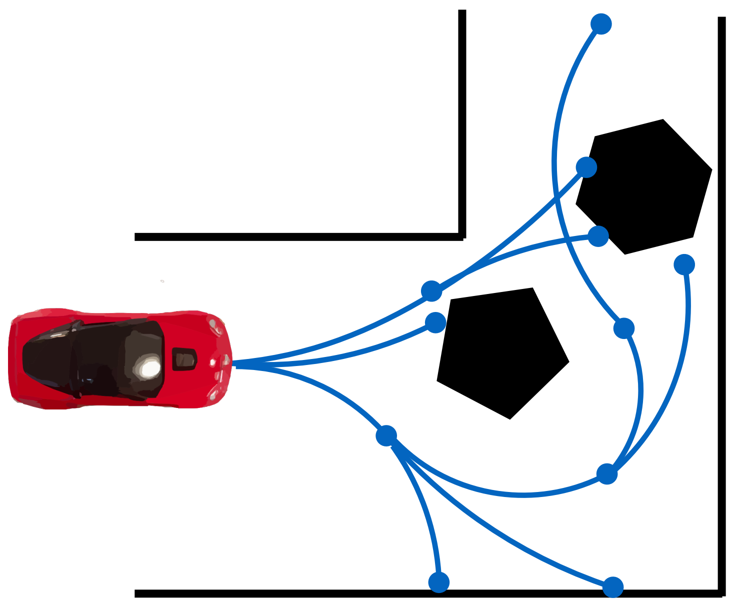
Class Overview

Estimate state



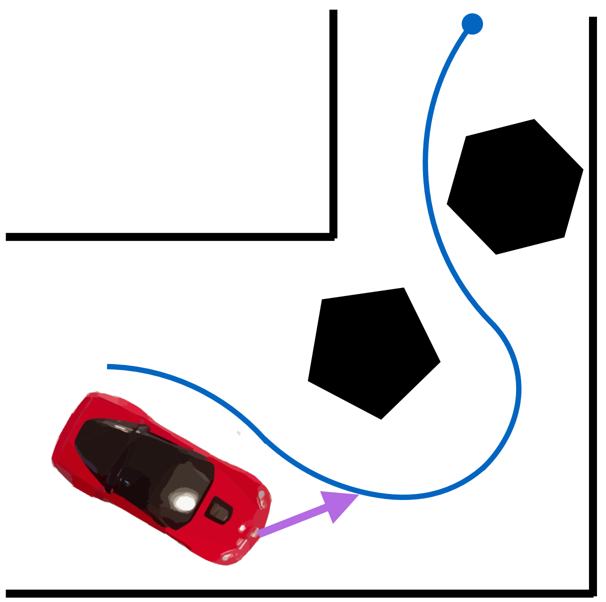
Weeks 2-4

Plan a sequence of motions



Weeks 7-8

Control robot to follow plan



Weeks 5-6