Partially known environment: Exploration and Safety

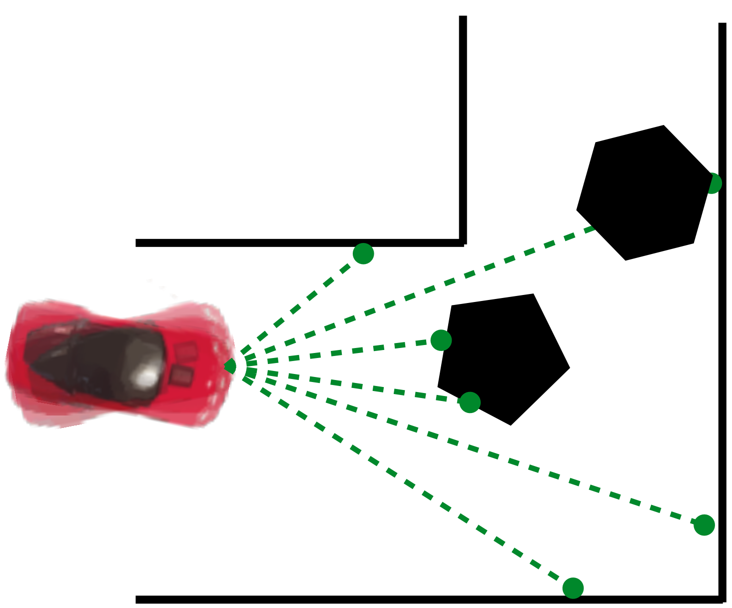
Sanjiban Choudhury

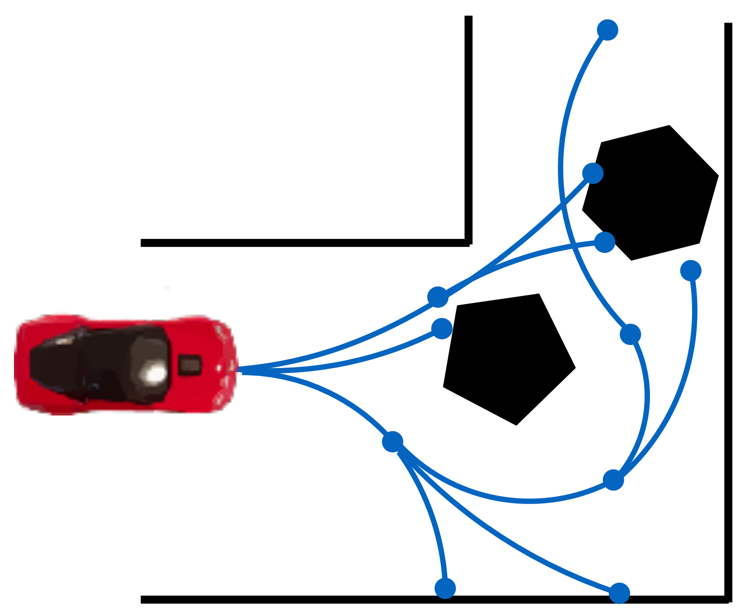
TAs: Matthew Rockett, Gilwoo Lee, Matt Schmittle

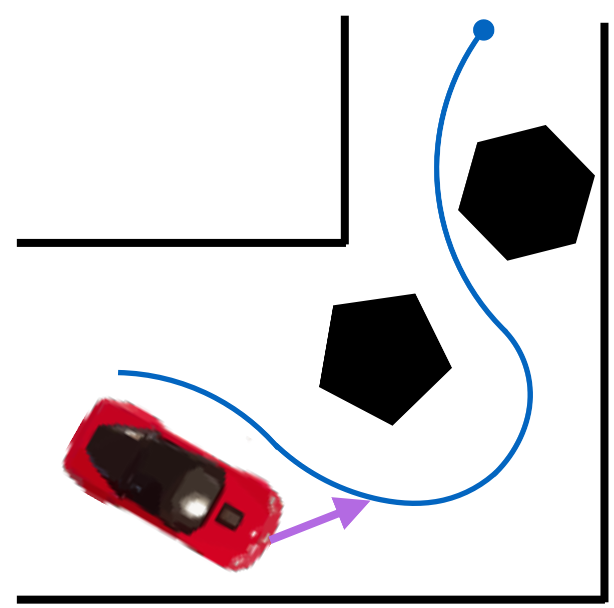
Estimate state

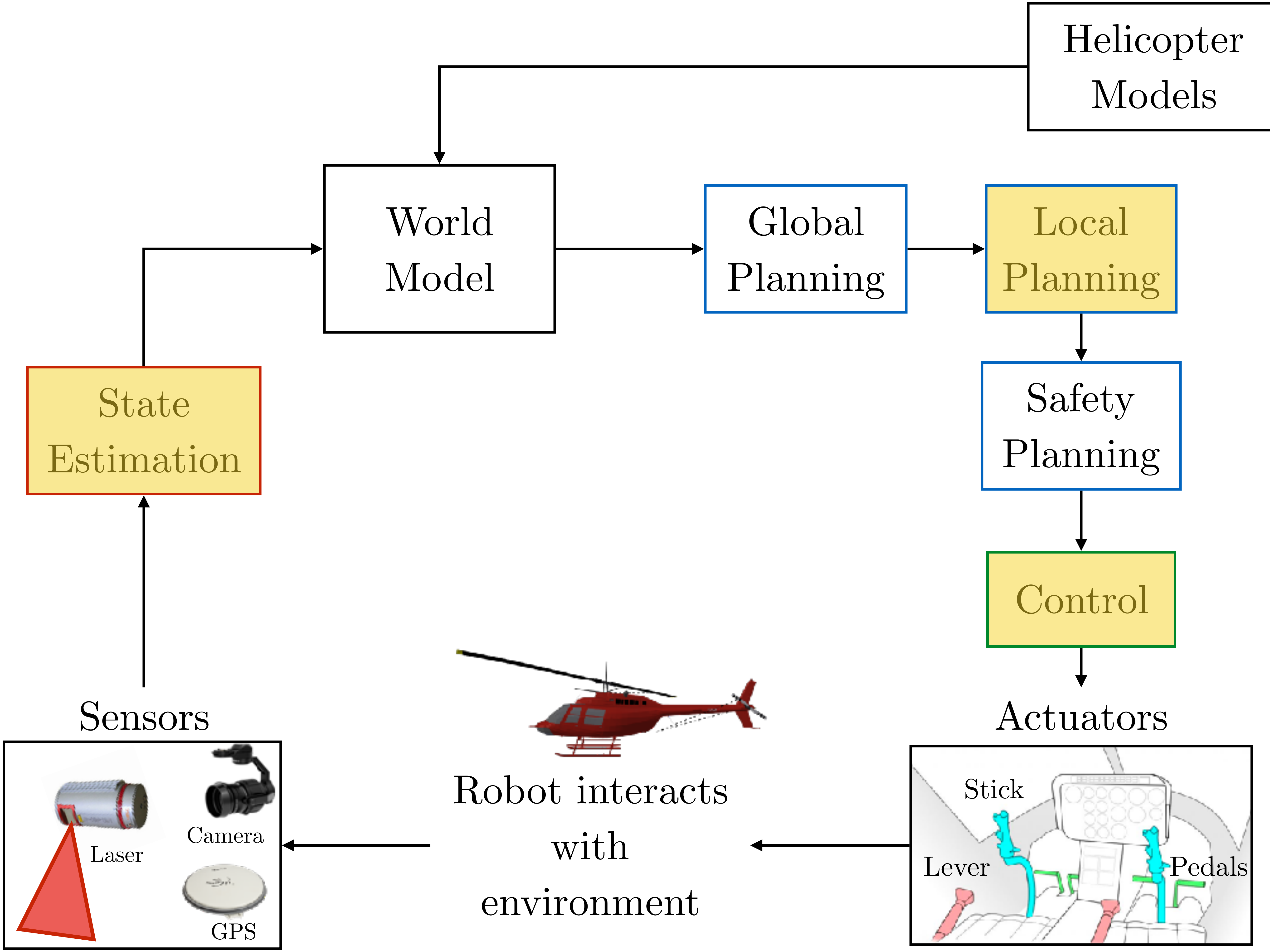
Plan a sequence of motions

Control robot to follow plan

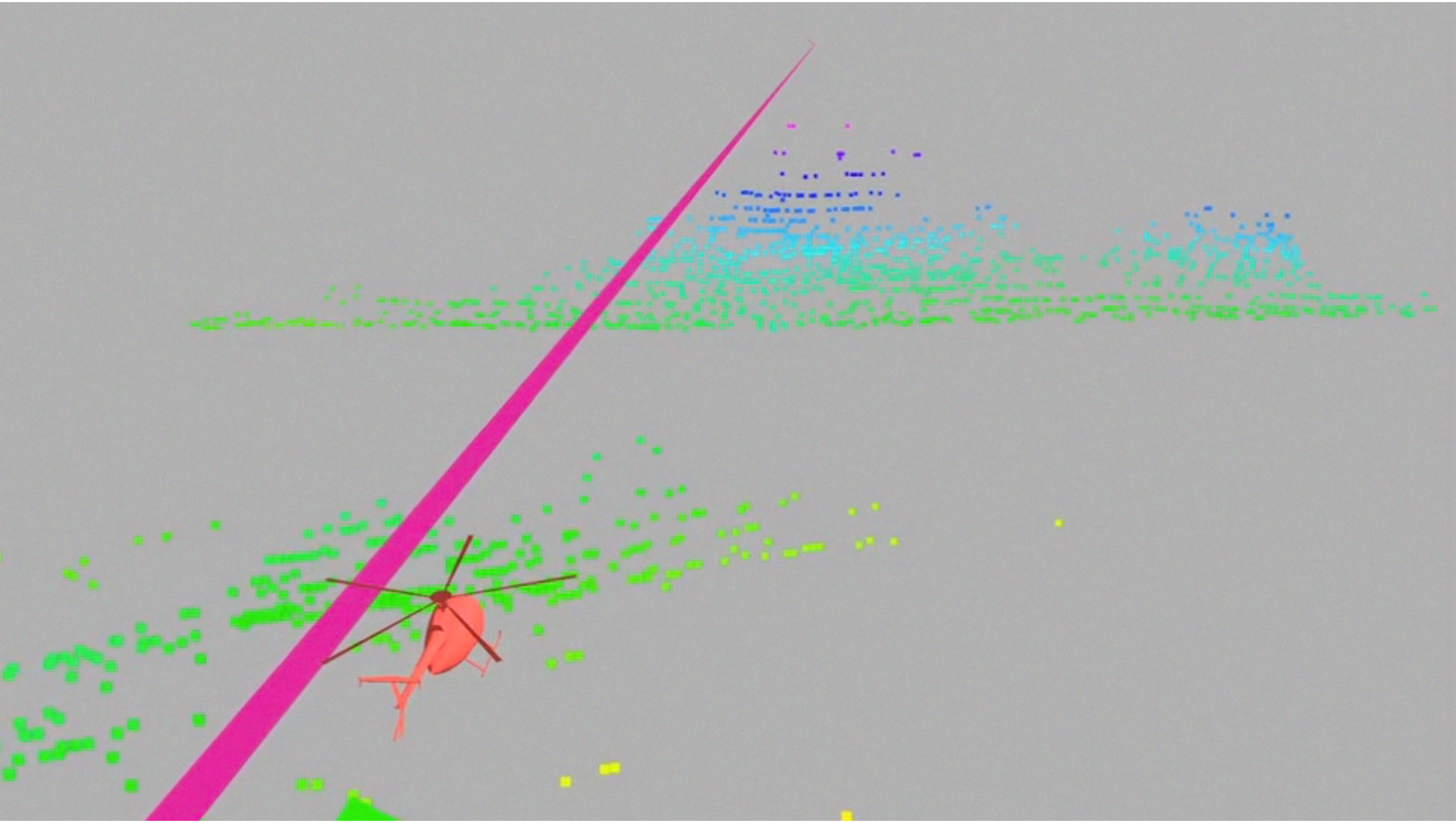







Helicopter Models
World Model
Global Planning
Local Planning
State Estimation
Safety Planning
Control
Sensors
Actuators
Laser
Camera
GPS
Robot interacts with environment
Stick
Lever
Pedals

Partially known environment



Motion Planning assumes the world is sufficiently known

What happens in a partially known world?

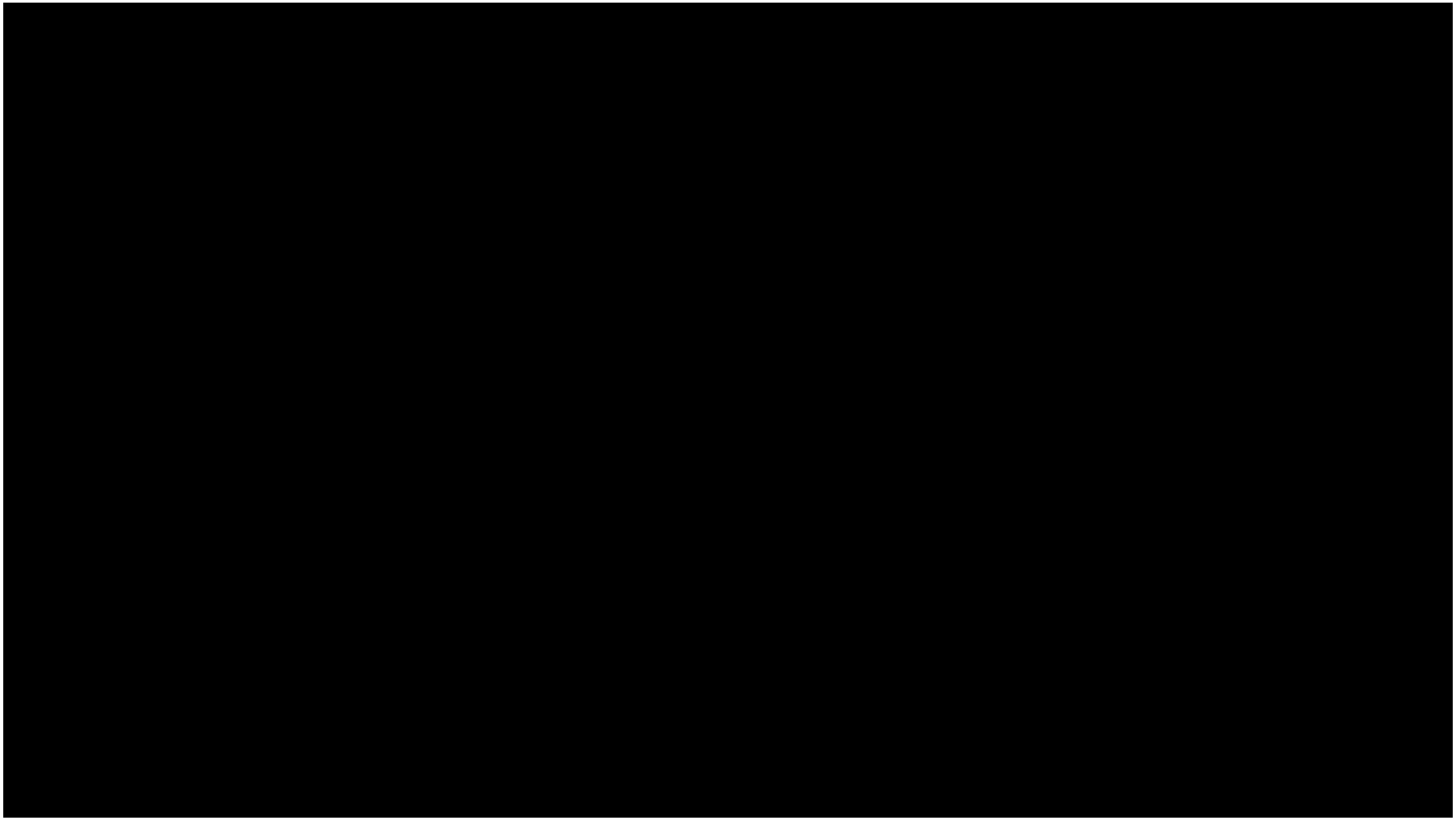
Two central questions

1. How do we gather information about the world?

2. How do we guarantee safety in a partially known world?

Information Gathering

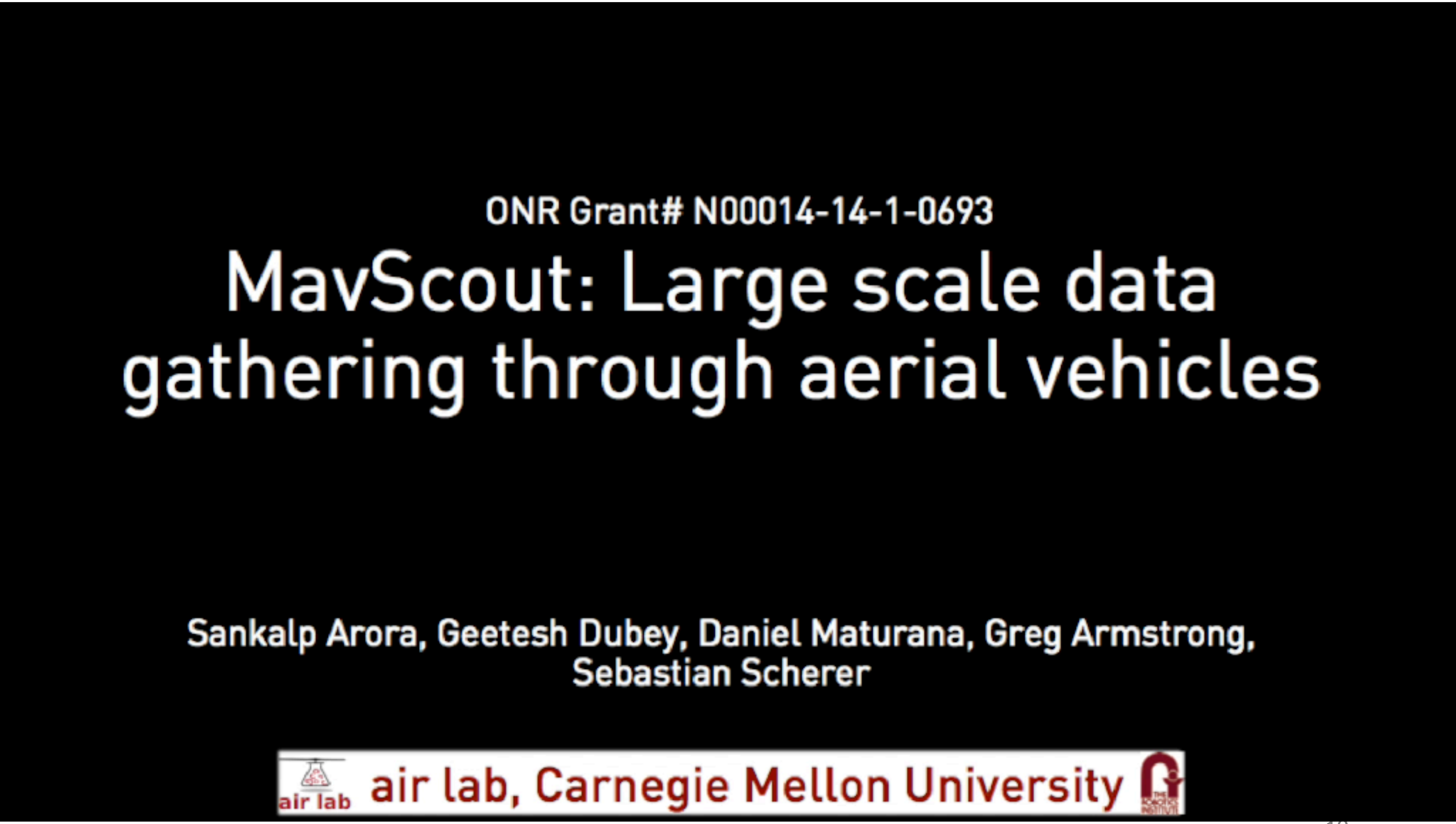
Autonomous indoor exploration



[1] Benjamin Charrow, Gregory Kahn, Sachin Patil, Sikang Liu, Ken Goldberg, Pieter Abbeel, Nathan Michael, and Vijay Kumar. Information-theoretic planning with trajectory optimization for dense 3d mapping. In RSS, 2015

Exploration of Subterranean Environment

Contextual Information Gathering



Informative Path Planning Problem

Plan a path to maximize the amount of information gathered
while respecting the total fuel constraints
and time constraints

What is information in this context?

Informative Path Planning Problem

Plan a path to maximize the amount of information gathered
while respecting the total fuel constraints
and time constraints

Let's look at a simpler problem

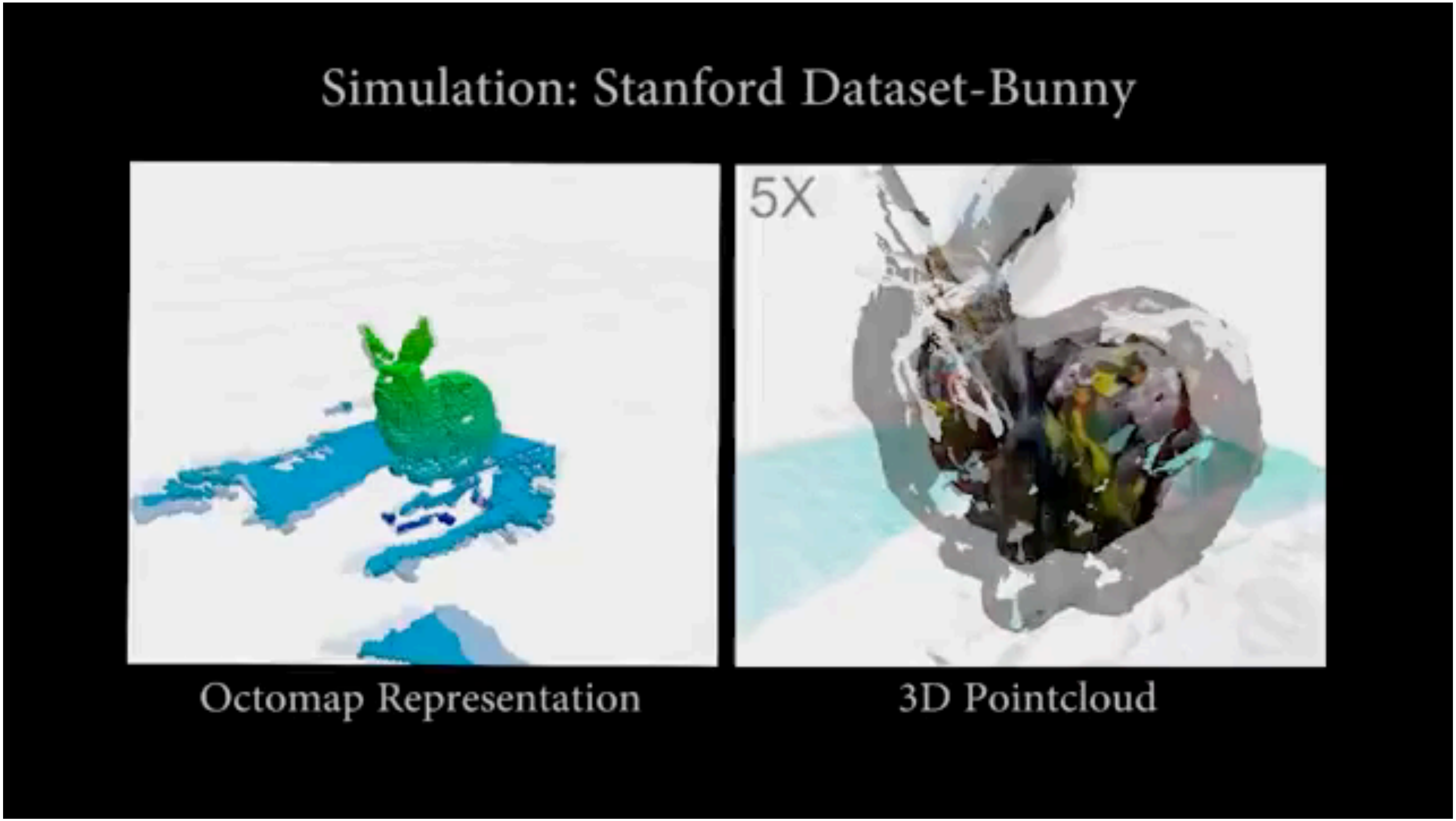
What if robot had infinite fuel

and

could teleport to any node?

Can we maximize the amount of information discovered by the robot?

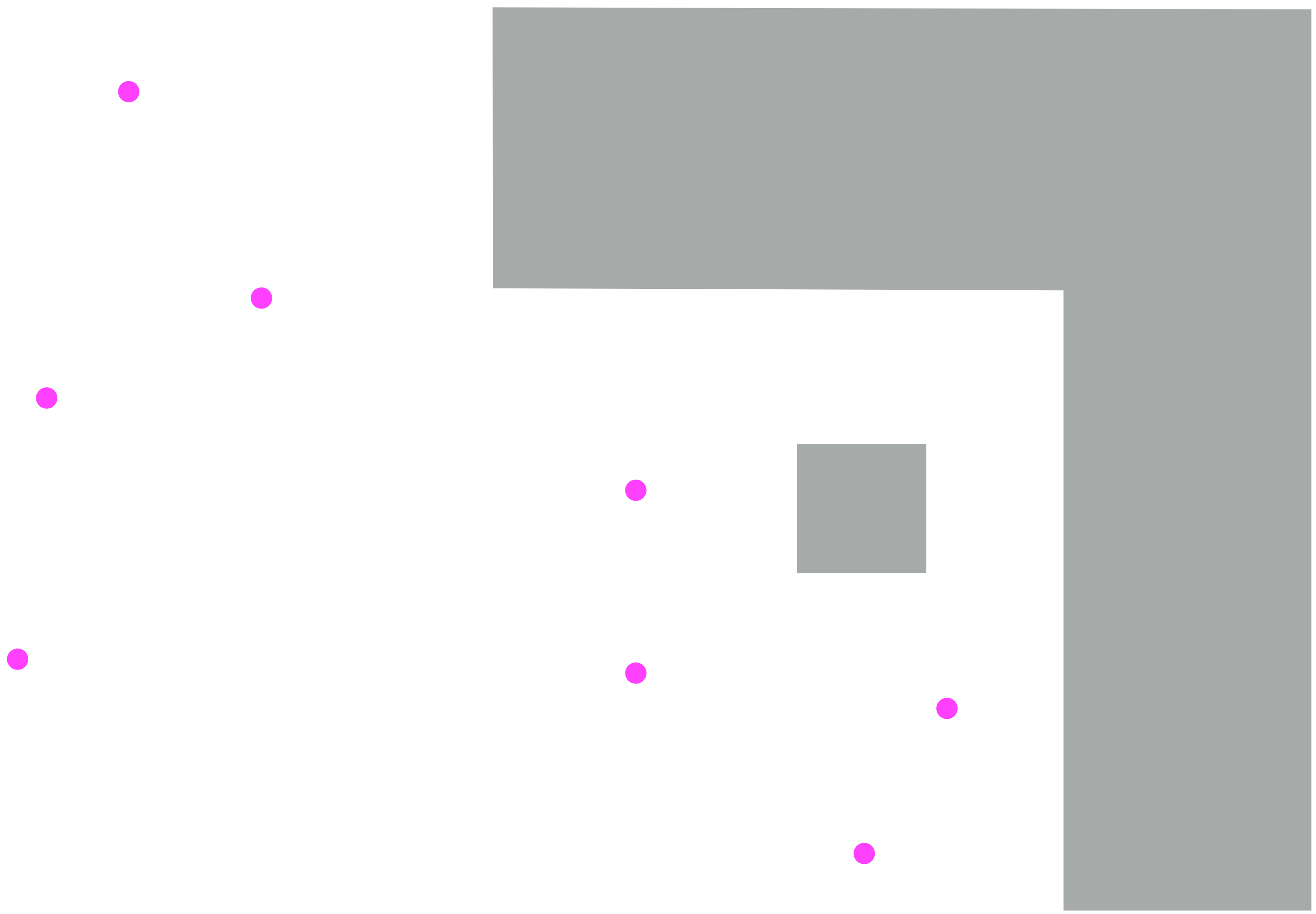
Sensor Placement Problem



[2] Stefan Isler, Reza Sabzevari, Jeffrey Delmerico, and Davide Scara- muzza. An information gain formulation for active volumetric 3d reconstruction. In ICRA, 2016.

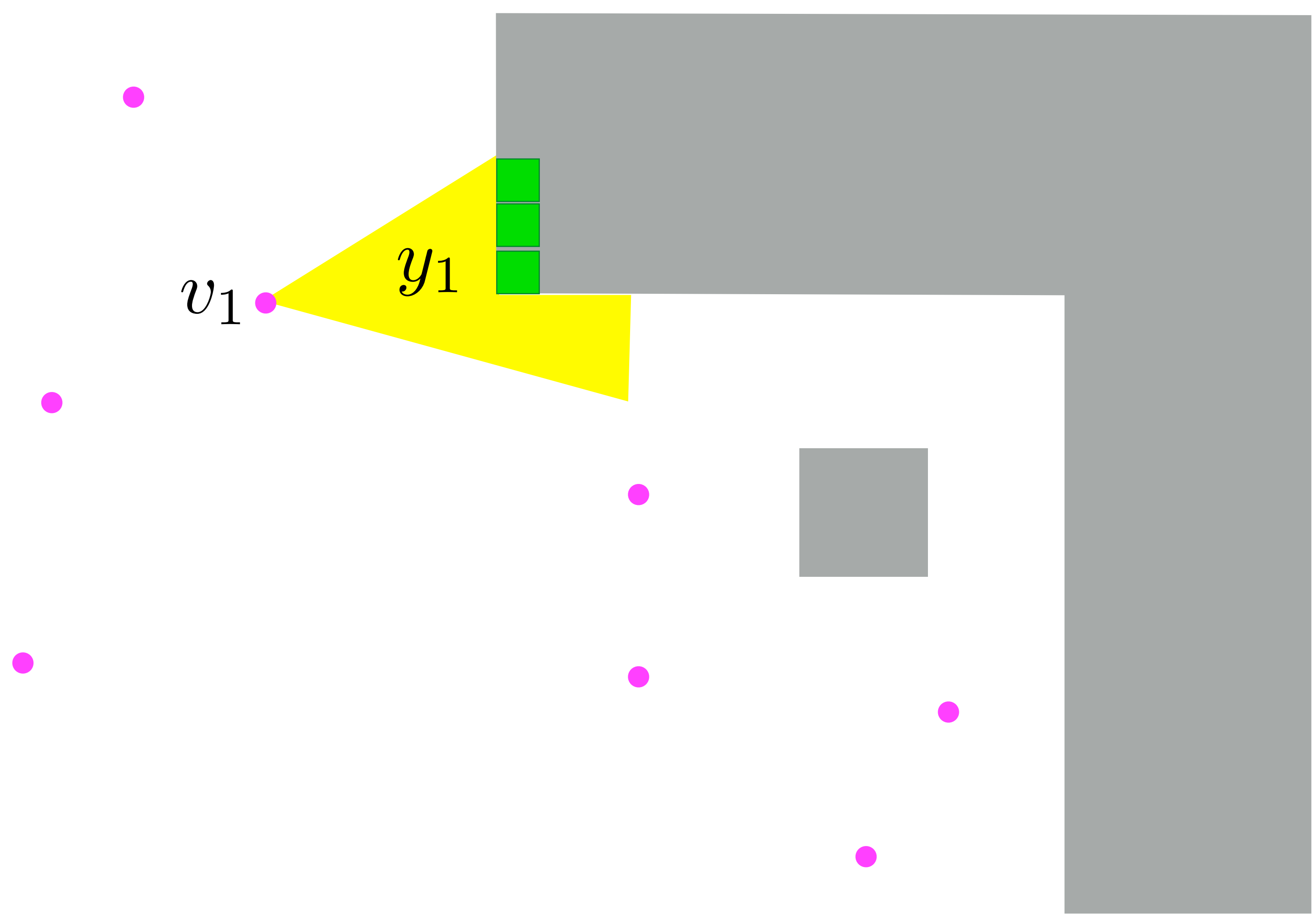
Sensor Placement Problem

$$\underset{\{v_1,\ldots,v_n\}}{\text{maximize}}\; F(v_1,\ldots,v_n)$$



Sensor Placement Problem

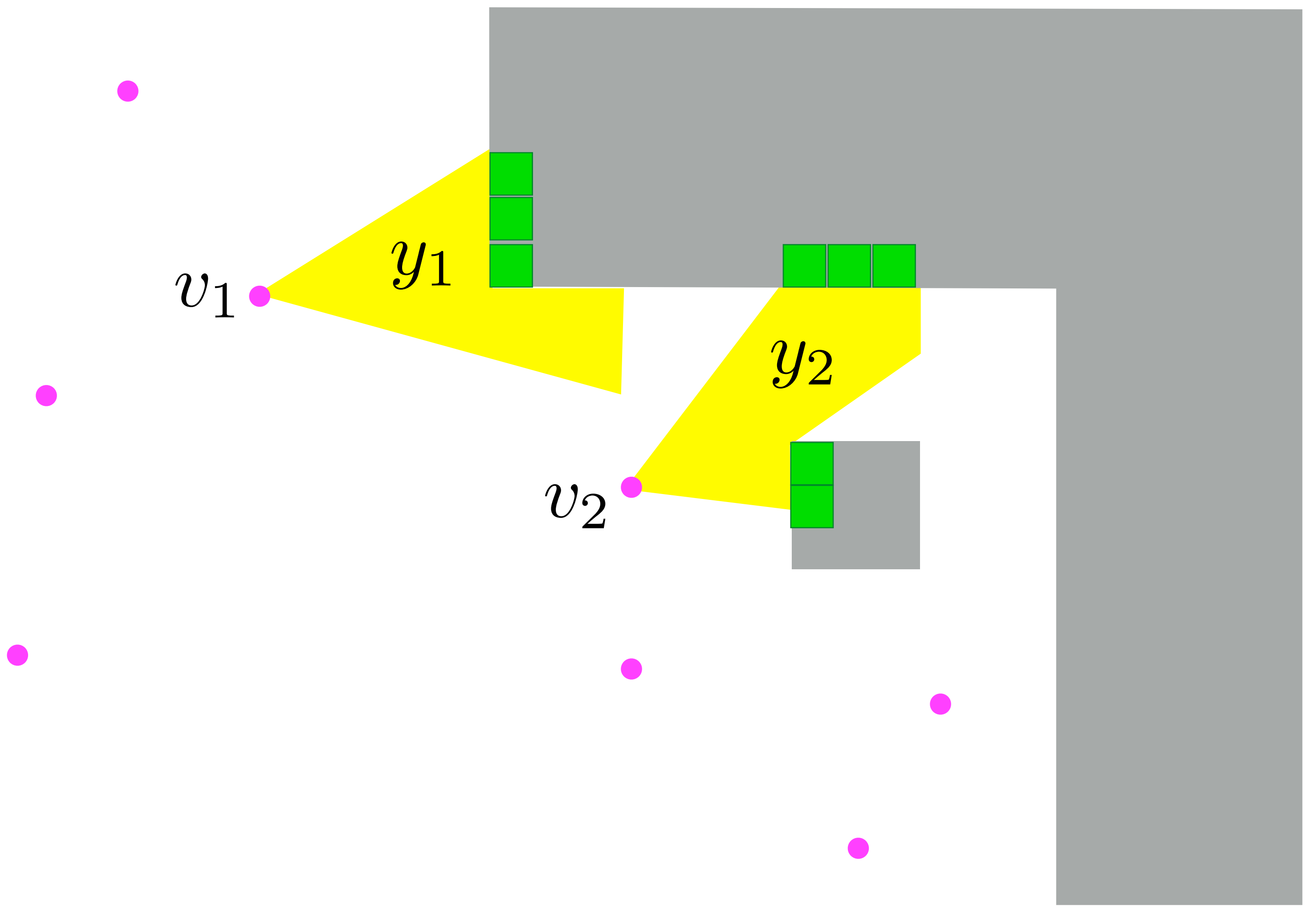
$$\underset{\{v_1,\ldots,v_n\}}{\text{maximize}}\; F(v_1,\ldots,v_n)$$



$F(v_1) = 3$

Sensor Placement Problem

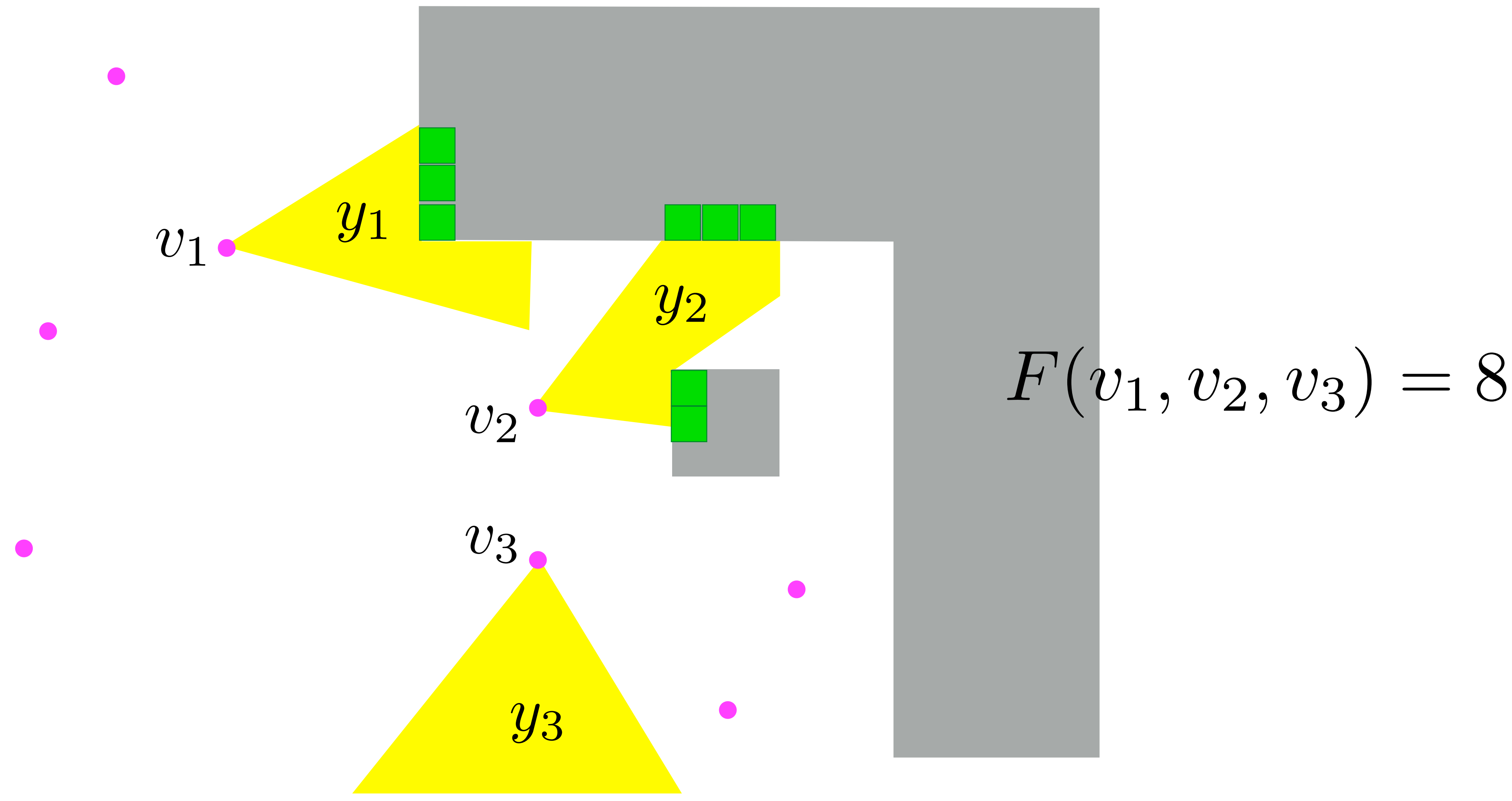
$$\underset{\{v_1,\ldots,v_n\}}{\text{maximize}} \; F(v_1,\ldots,v_n)$$



$F(v_1, v_2) = 8$

Sensor Placement Problem

$$\underset{\{v_1,\ldots,v_n\}}{\text{maximize}}\; F(v_1,\ldots,v_n)$$



Can't we run dynamic programming and get the optimal answer?

No! Lack of optimal substructure

Optimal substructure in Search

$$g(s) = \min_{s' \in \mathrm{pred}(s)} \left(g(s') + c(s', s) \right)$$

Optimal substructure in LQR

Recall the Bellman function that relates value at consecutive time steps

$$J(x_t, t) = \min_{u_t} c(x_t, u_t) + J(x_{t+1}, t+1)$$
$$= \min_{u_t} x_t^T Q x_t + u_t^T R u_t + J(x_{t+1}, t+1)$$

No optimal substructure in IPP

$$F(v_1, \ldots, v_n)$$

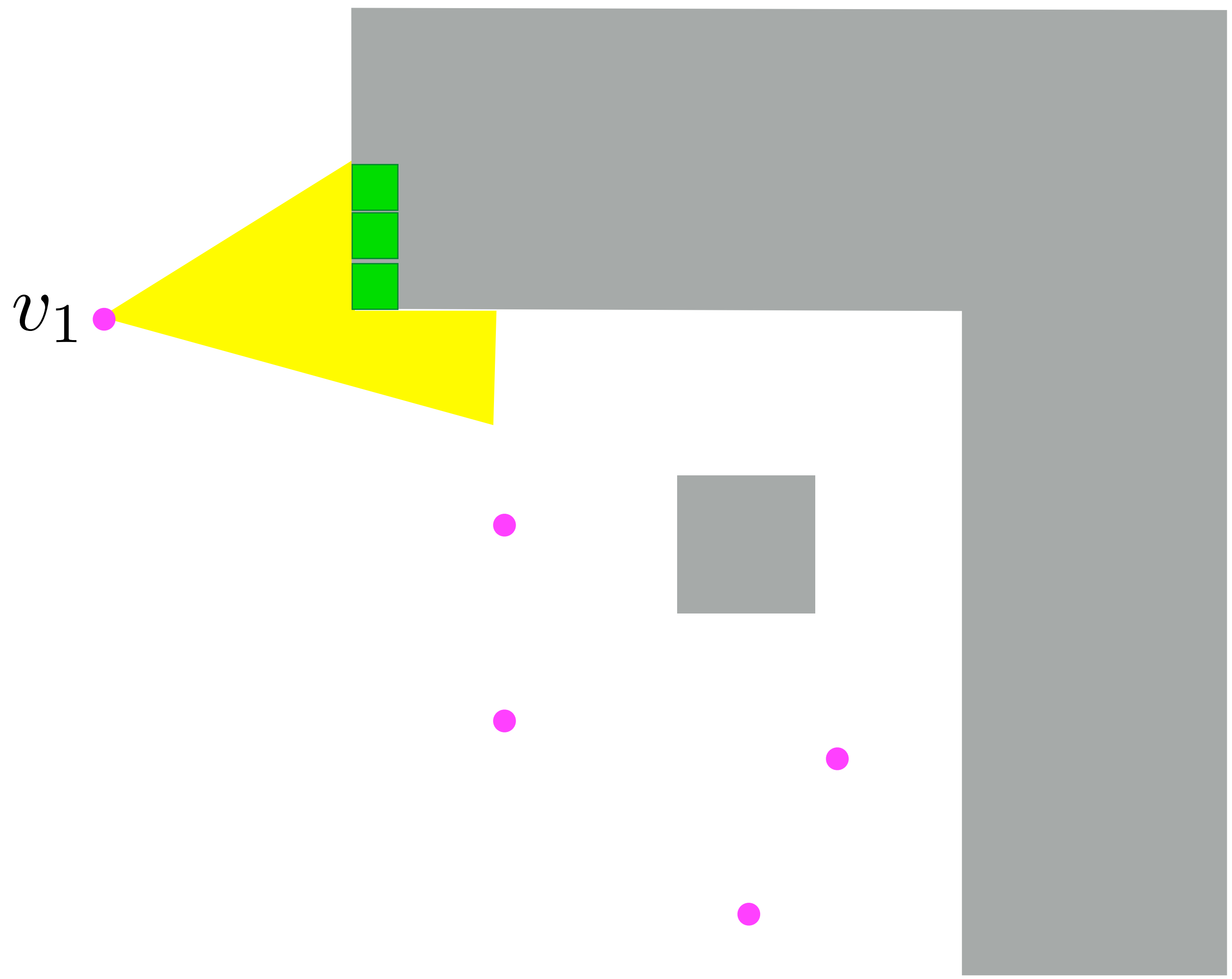
Here F is a set function.

The utility of adding a vertex depends on the vertices already added.

Utility is a set function

v_1 increases the utility by 3; seems informative

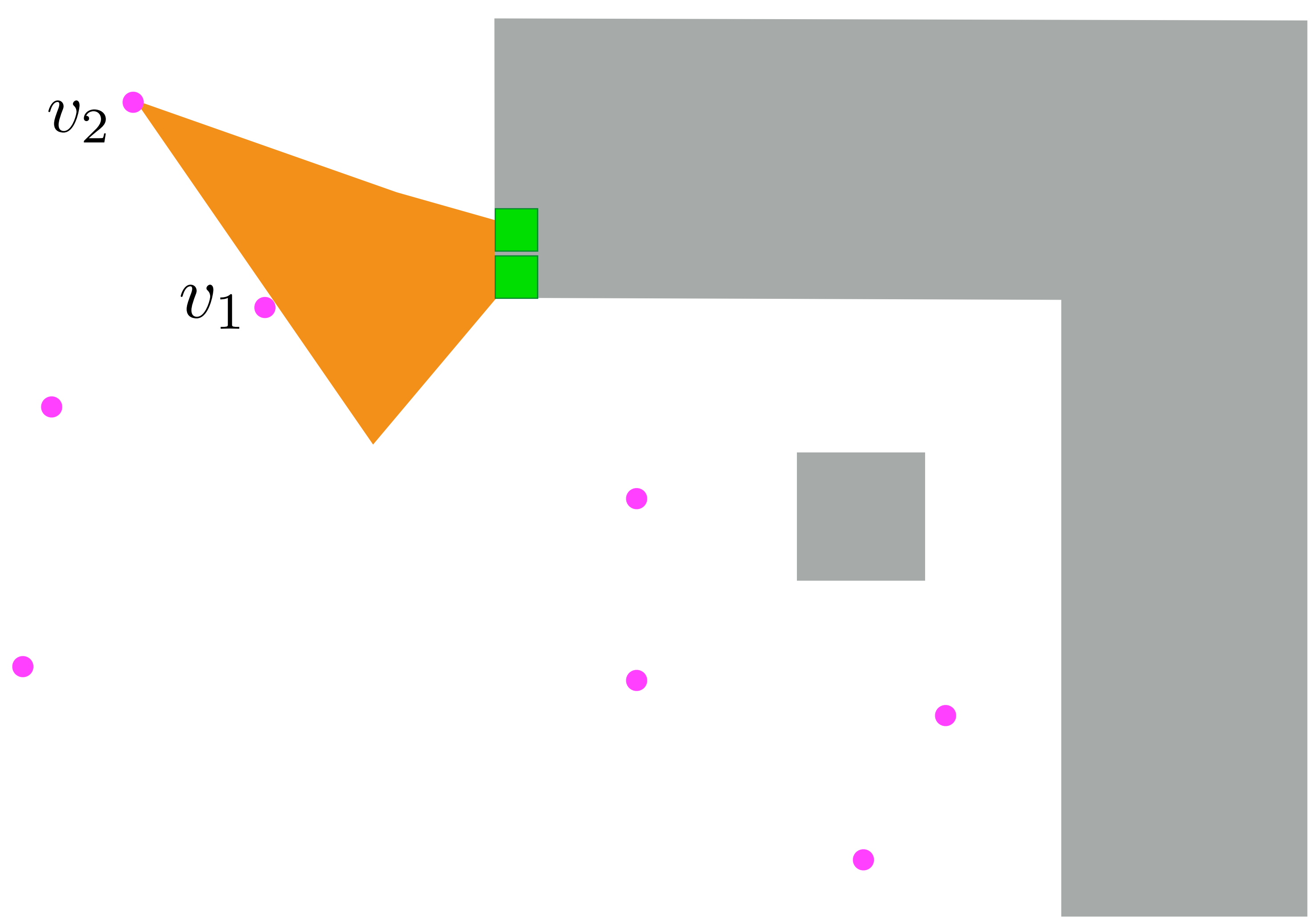
$$F(\emptyset) = 0 \quad F(v_1) = 3$$



Utility is a set function

Now let's say we have already visited v_2

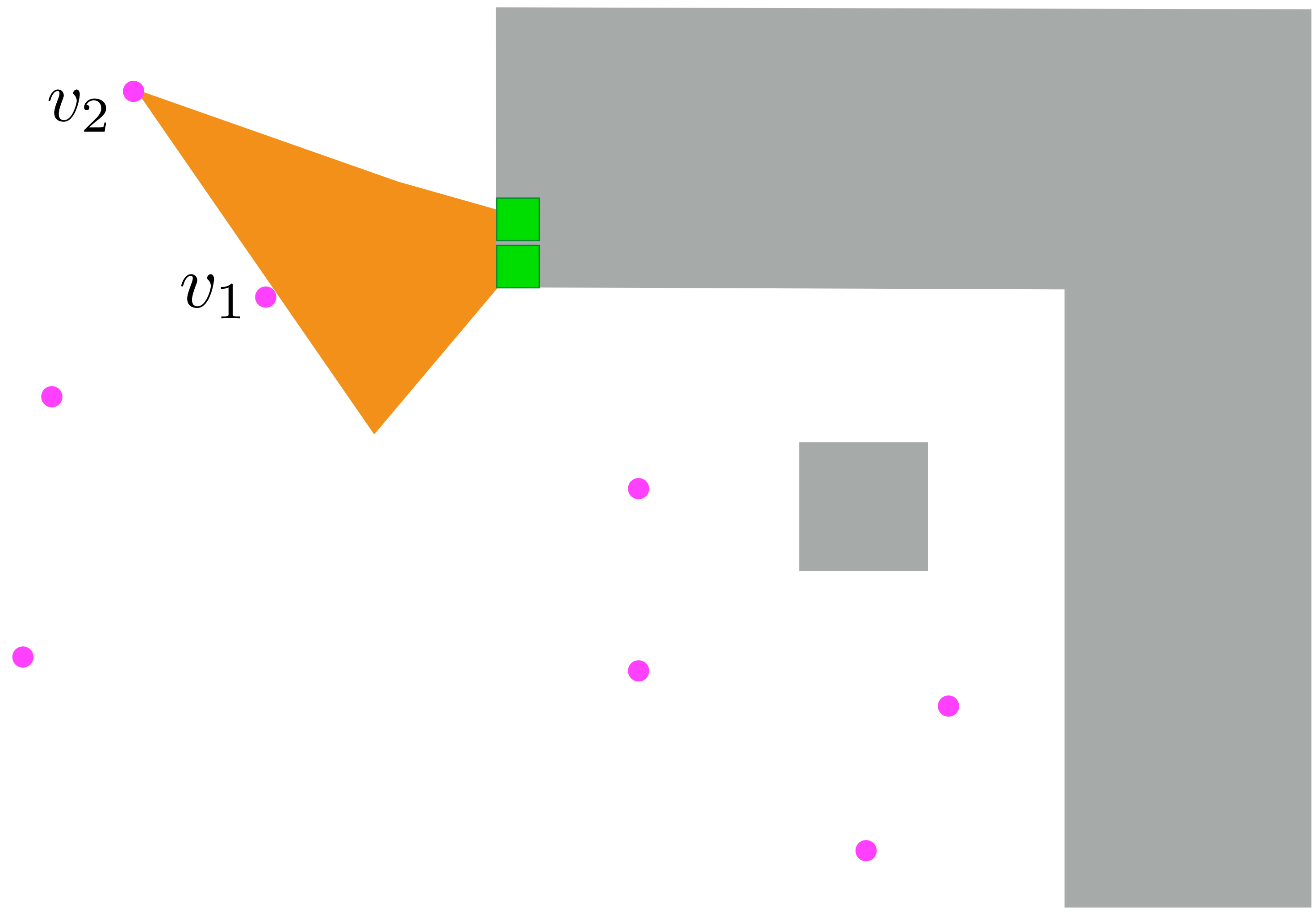
$$F(v_2) = 2$$



Utility is a set function

Additional contribution of v_1 is only 1

$$F(v_2) = 2 \qquad F(v_2, v_1) = 3$$



Utility of a vertex depends on the path we took to get there

Need to reason over all combination of paths!

NP-hard

The provable virtue of greediness

Submodular Functions

Submodularity is a property of *set functions*, i.e., functions $f : 2^V \to \mathbb{R}$

Definition 1.1 (Discrete derivative) For a set function $f : 2^V \to \mathbb{R}$, $S \subseteq V$, and $e \in V$, let $\Delta_f(e \mid S) := f(S \cup \{e\}) - f(S)$ be the *discrete derivative* of f at S with respect to e.

Where the function f is clear from the context, we drop the subscript and simply write $\Delta(e \mid S)$.

Submodular Functions

Definition 1.2 (Submodularity) A function $f : 2^V \to \mathbb{R}$ is *submodular* if for every $A \subseteq B \subseteq V$ and $e \in V \setminus B$ it holds that

$$\Delta(e \mid A) \geq \Delta(e \mid B)\,.$$

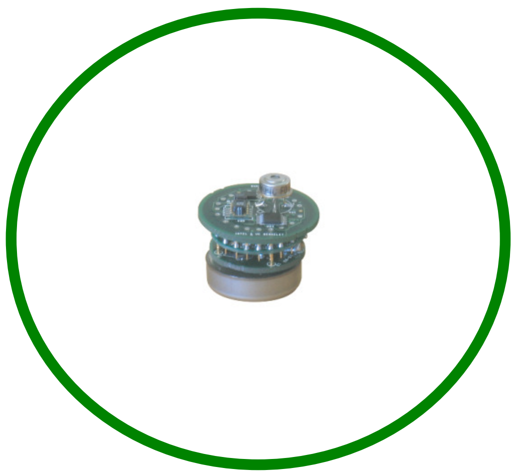
(think of this as diminishing returns)

Definition 1.3 (Monotonicity) A function $f : 2^V \to \mathbb{R}$ is *monotone* if for every $A \subseteq B \subseteq V$, $f(A) \leq f(B)$.

(think of this as positive returns)

Example: Set cover

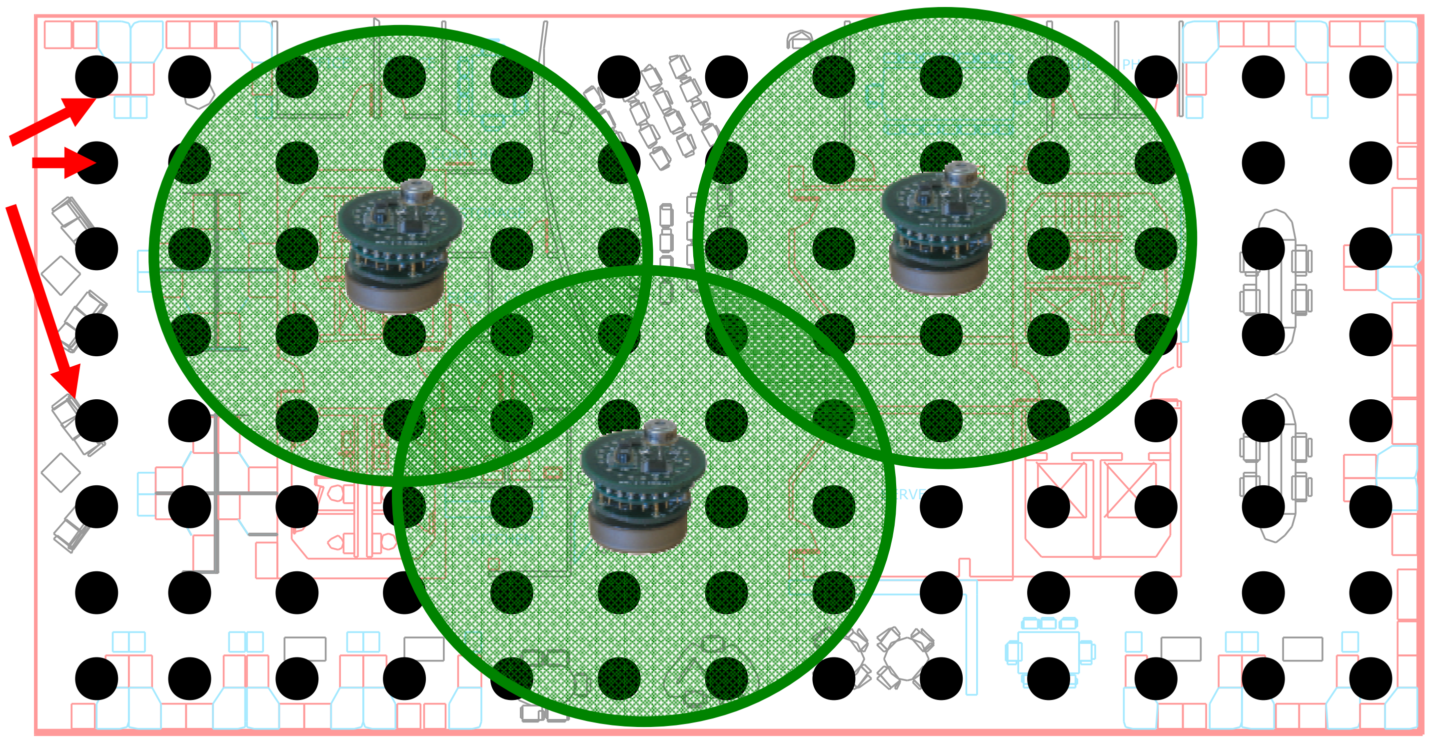
Place sensors in building



Node predicts values of positions with some radius

Want to cover floorplan with discs

Possible locations V



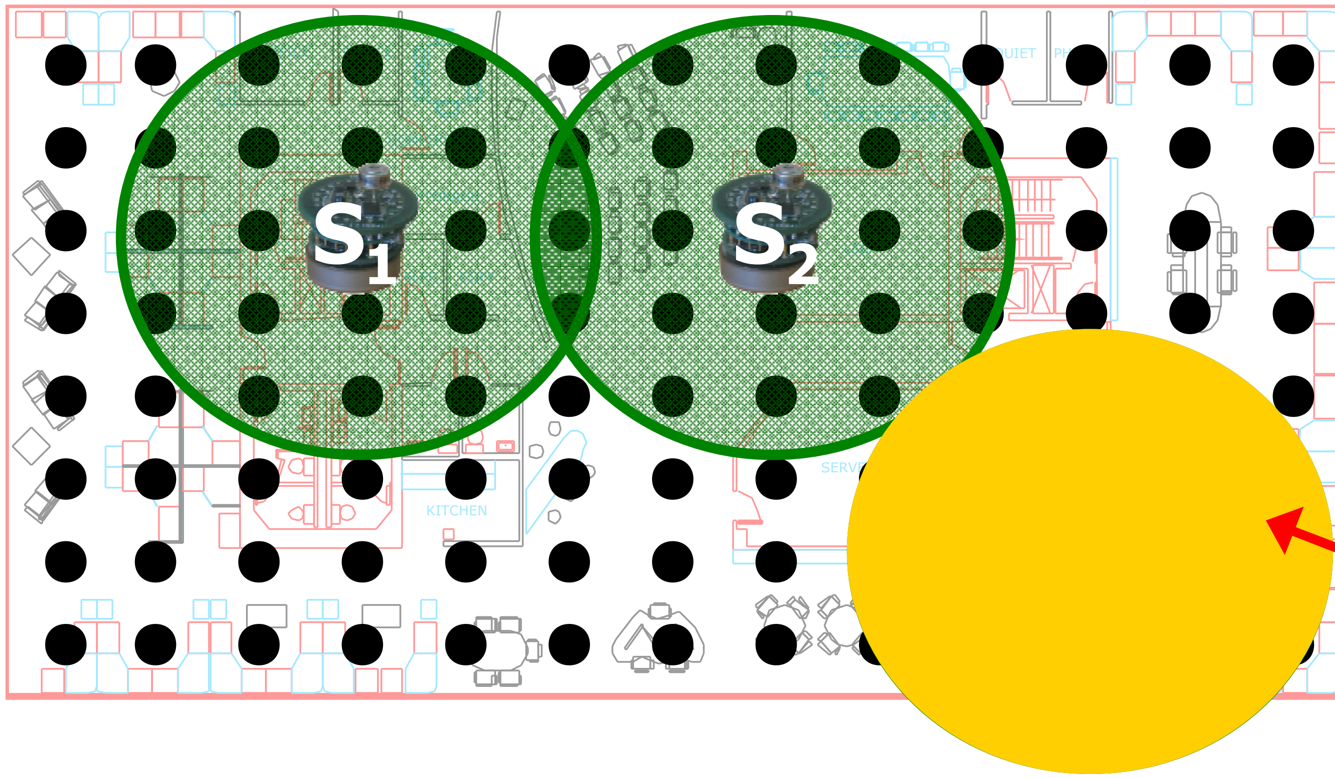
For $A \subseteq V$: F(A) = "area covered by sensors placed at A"

Formally:

W finite set, collection of n subsets $S_i \subseteq W$

For $A \subseteq V=\{1,\dots,n\}$ define $F(A) = |\bigcup_{i \in A} S_i|$

Set cover is submodular

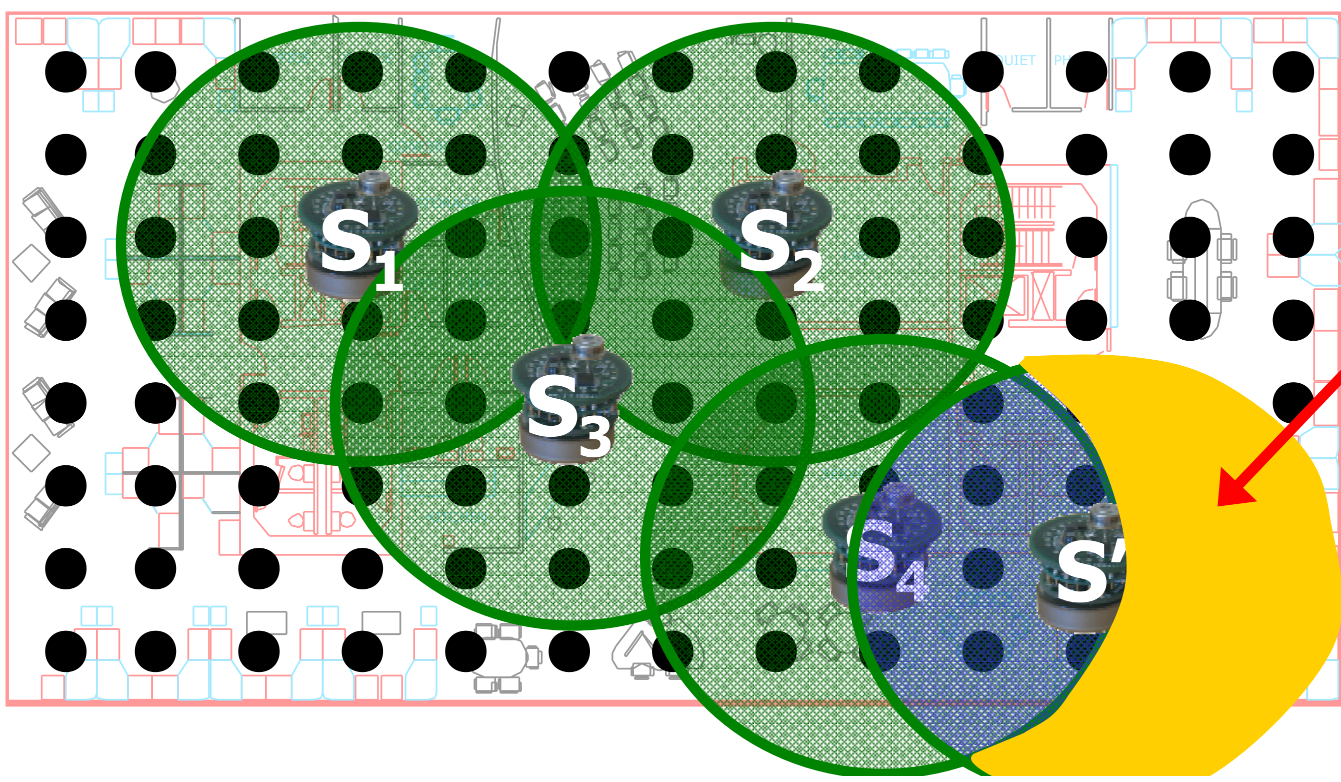


$A=\{S_1,S_2\}$

$F(A\cup\{S'\})-F(A)$

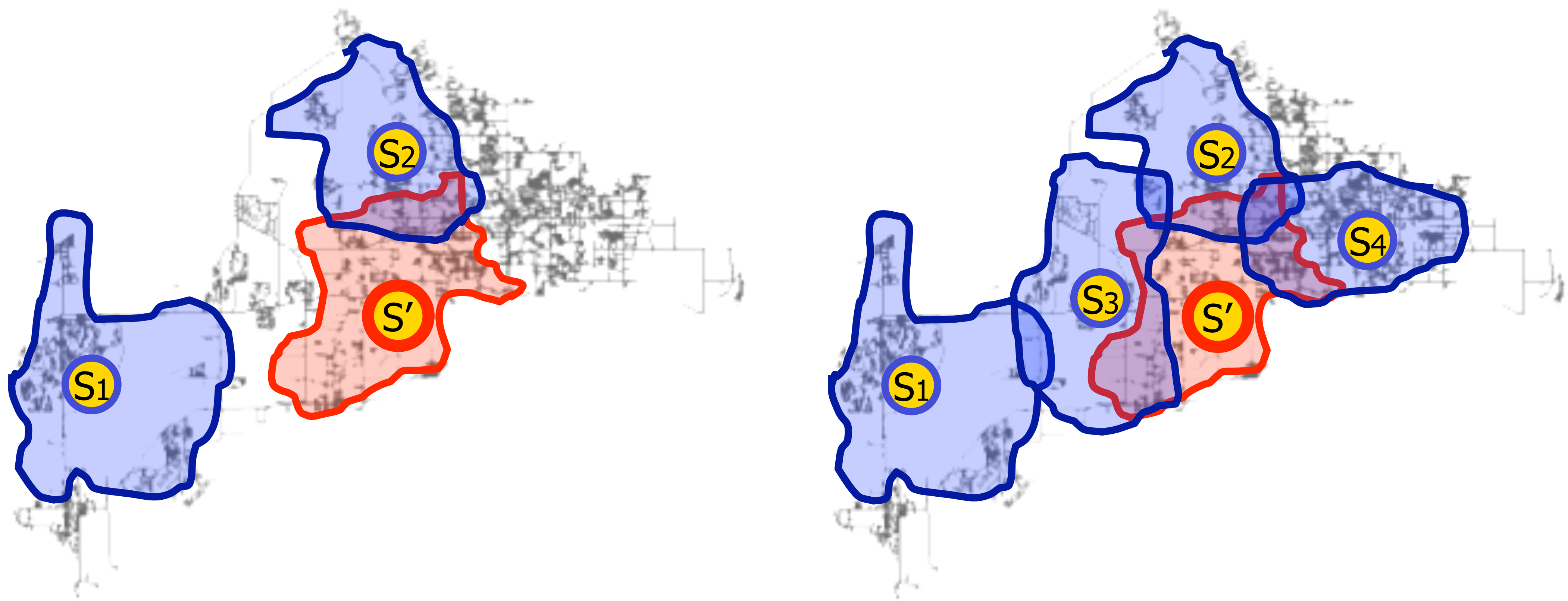
$\geq$

$F(B\cup\{S'\})-F(B)$



$B = \{S_1,S_2,S_3,S_4\}$

Example: Sensor placement



(a) *Adding* s' *to set* $\{s_1, s_2\}$ (b) *Adding* s' *to superset* $\{s_1, \ldots, s_4\}$

Figure 1 Illustration of the diminishing returns effect in context of placing sensors in a water distribution network to detect contaminations. The blue regions indicate nodes where contamination is detected quickly using the existing sensors S. The red region indicates the additional coverage by adding a new sensor s'. If more sensors are already placed (b), there is more overlap, hence less gain in utility: $\Delta(s' \mid \{s_1, s_2\}) \geq \Delta(s' \mid \{s_1, \ldots, s_4\})$.

Theorem: Greedy is near-optimal

$$S_i = S_{i-1} \cup \{\arg\max_e \Delta(e \mid S_{i-1})\}.$$

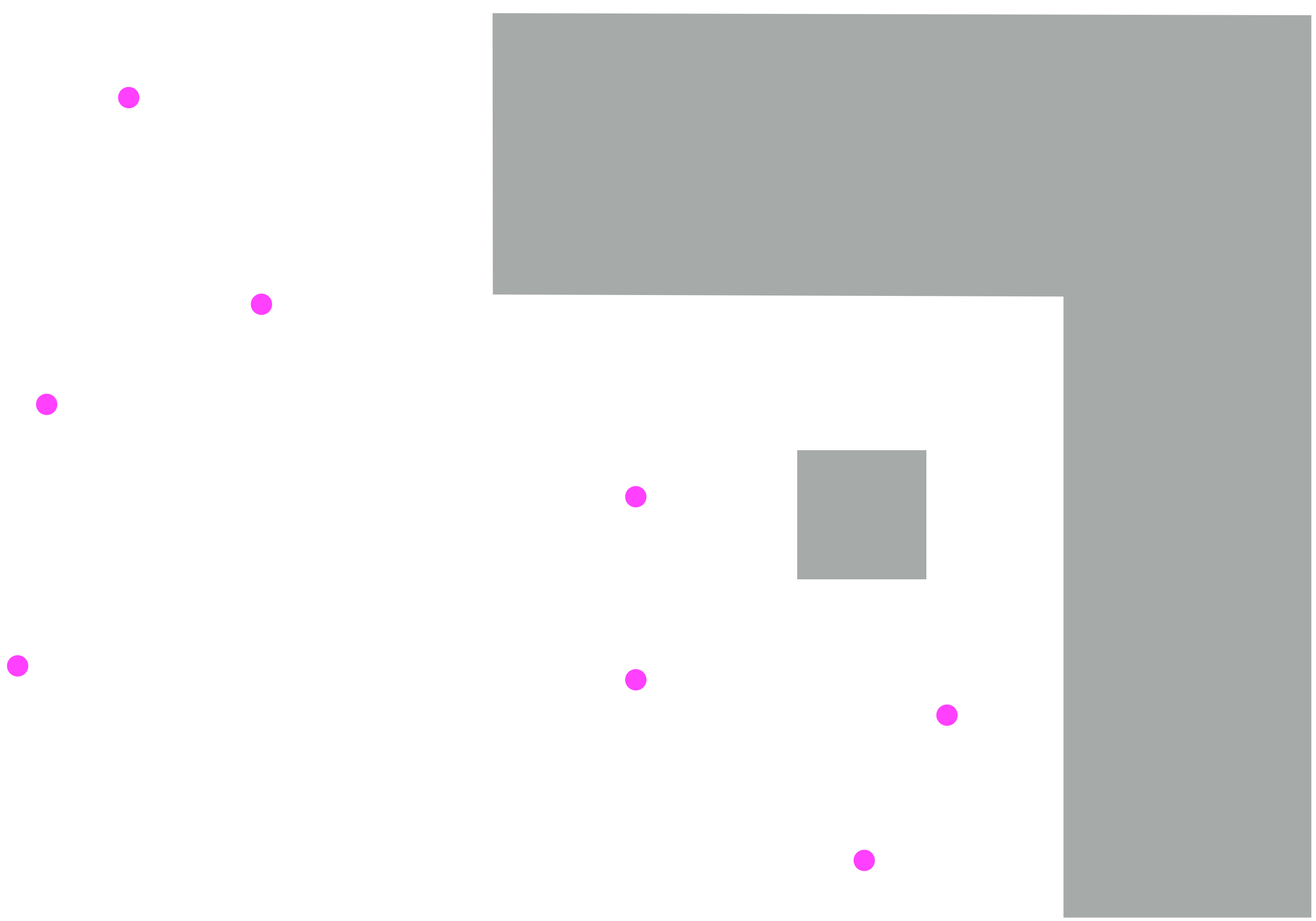
Theorem 1.5 (Nemhauser et al. 1978) *Fix a nonnegative monotone submodular function* $f : 2^V \to \mathbb{R}_+$ *and let* $\{S_i\}_{i \geq 0}$ *be the greedily selected sets defined in Eq. (2). Then for all*

$$f(S_k) \geq \left(1 - \frac{1}{e}\right) f(S_k^*)$$

(greedy) 63% (optimal)

Back to our problem ...

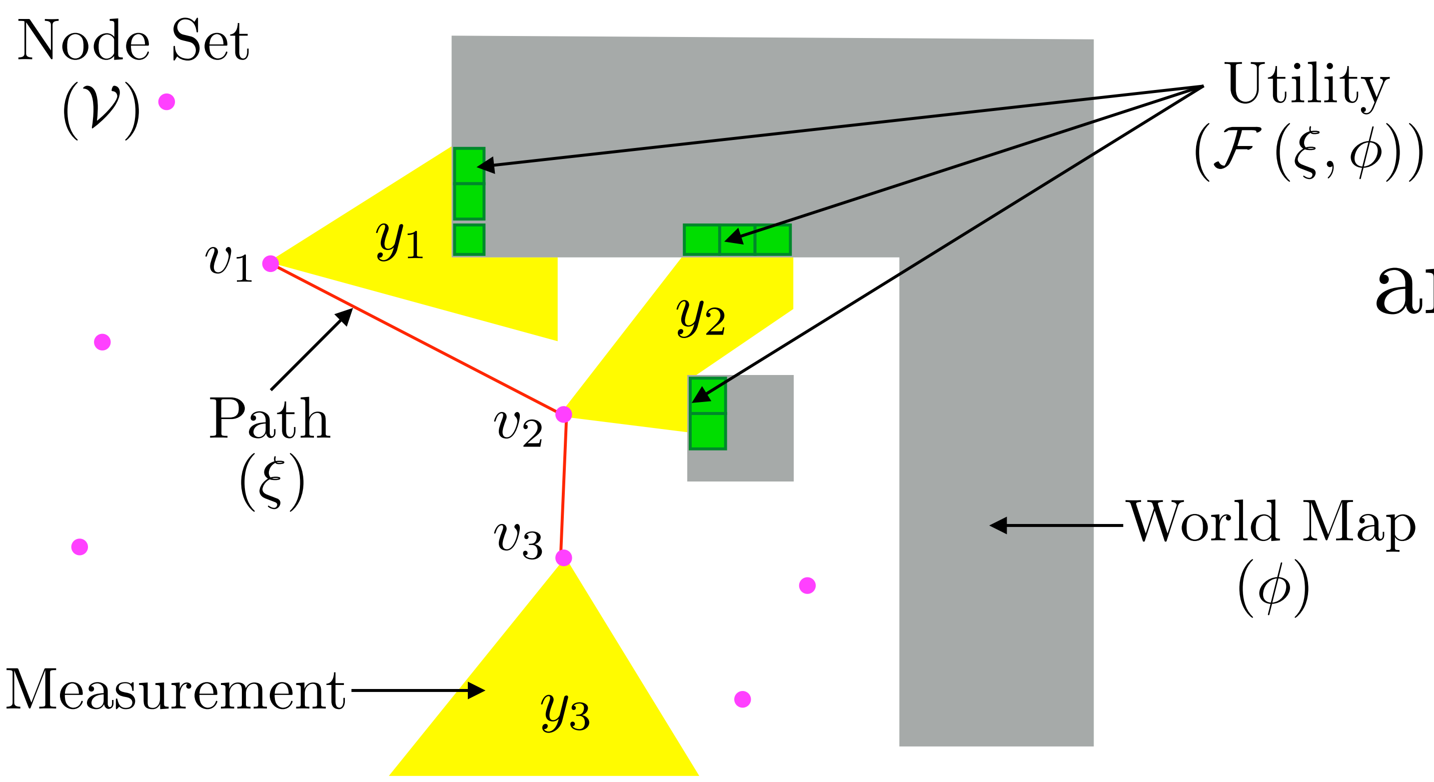
$$\underset{\{v_1,\ldots,v_n\}}{\text{maximize}}\ F(v_1,\ldots,v_n)$$



Greedily visit nodes with highest marginal utility

Informative Path Planning Problem

What if we could no longer teleport?



$$\arg\max_{\xi \in \Xi} \quad \mathcal{F}(\xi, \phi) \qquad \text{(Coverage)}$$

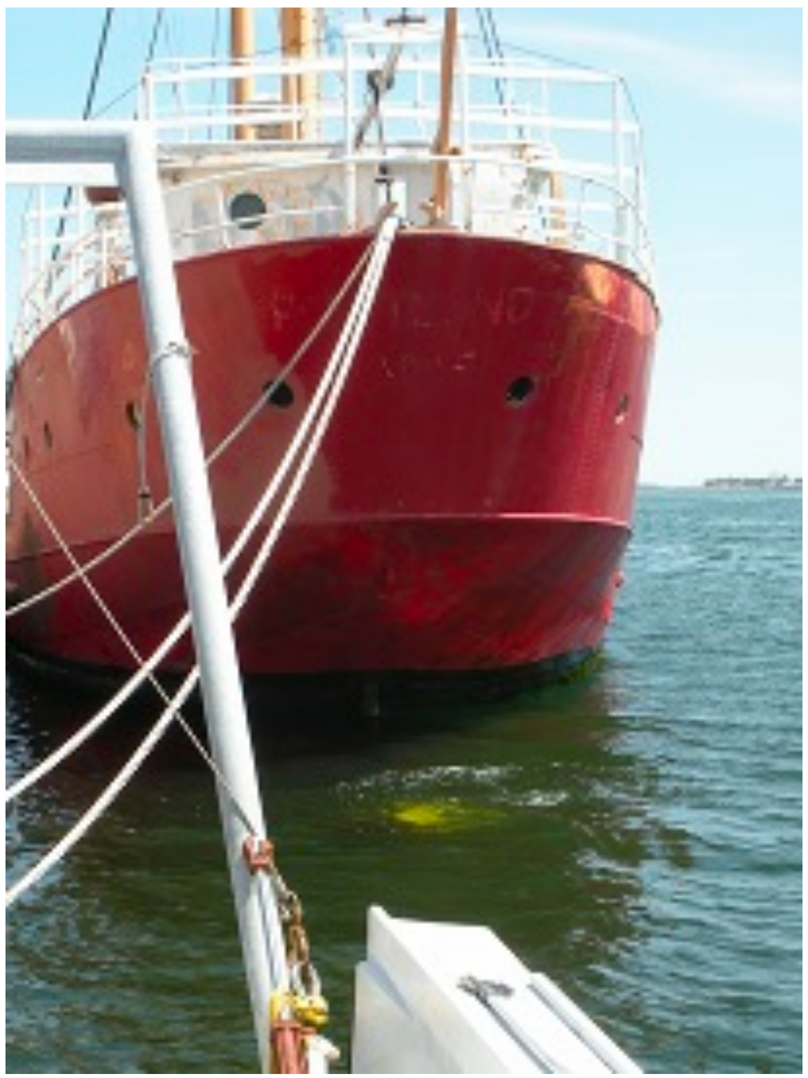
$$s.t. \quad \mathcal{T}(\xi, \phi) \leq B \qquad \text{(Tour Cost)}$$

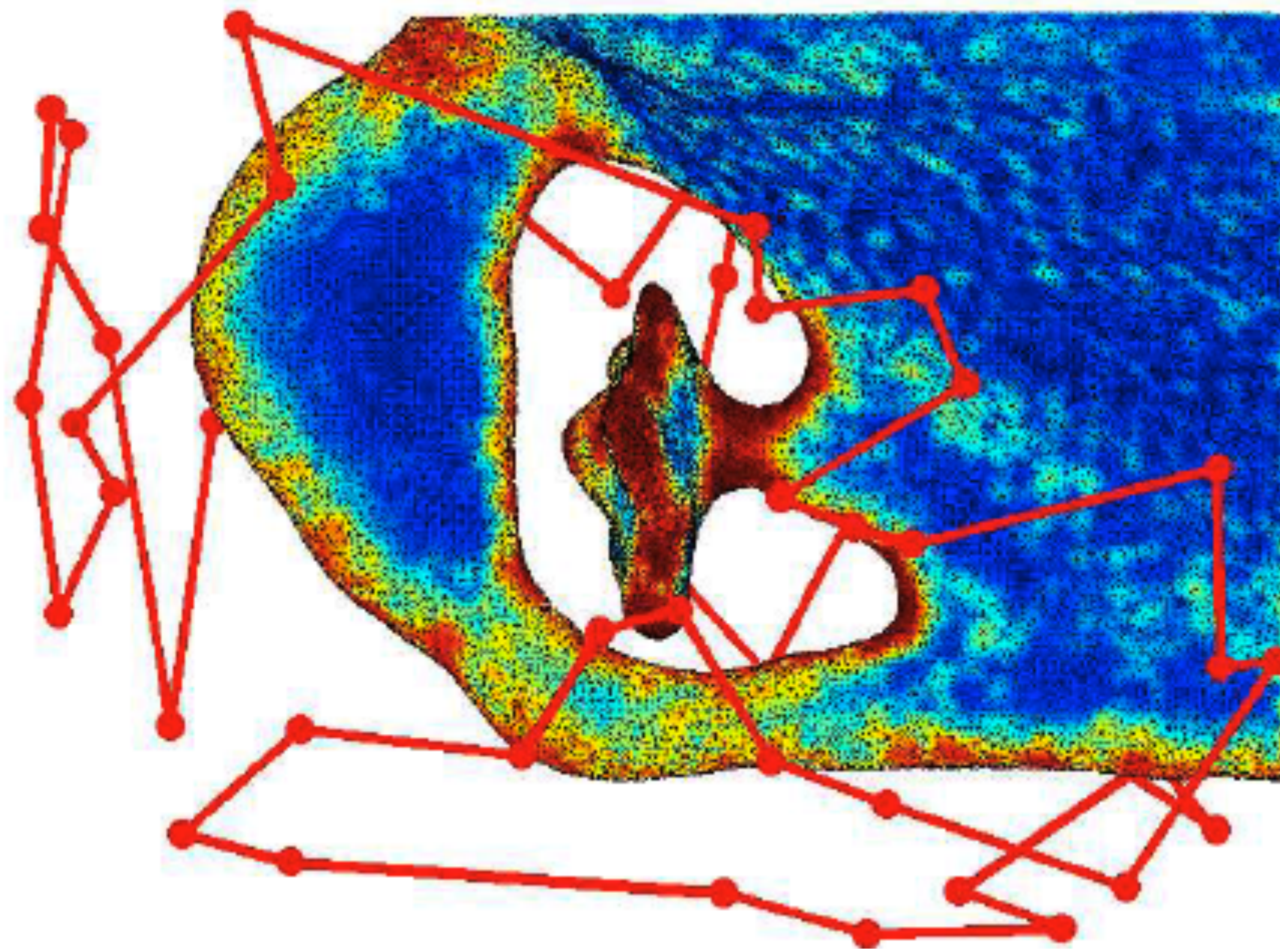
Would greedy still be near-optimal?

Generalized Cost Benefit

(on board)

Ship Hull Inspection

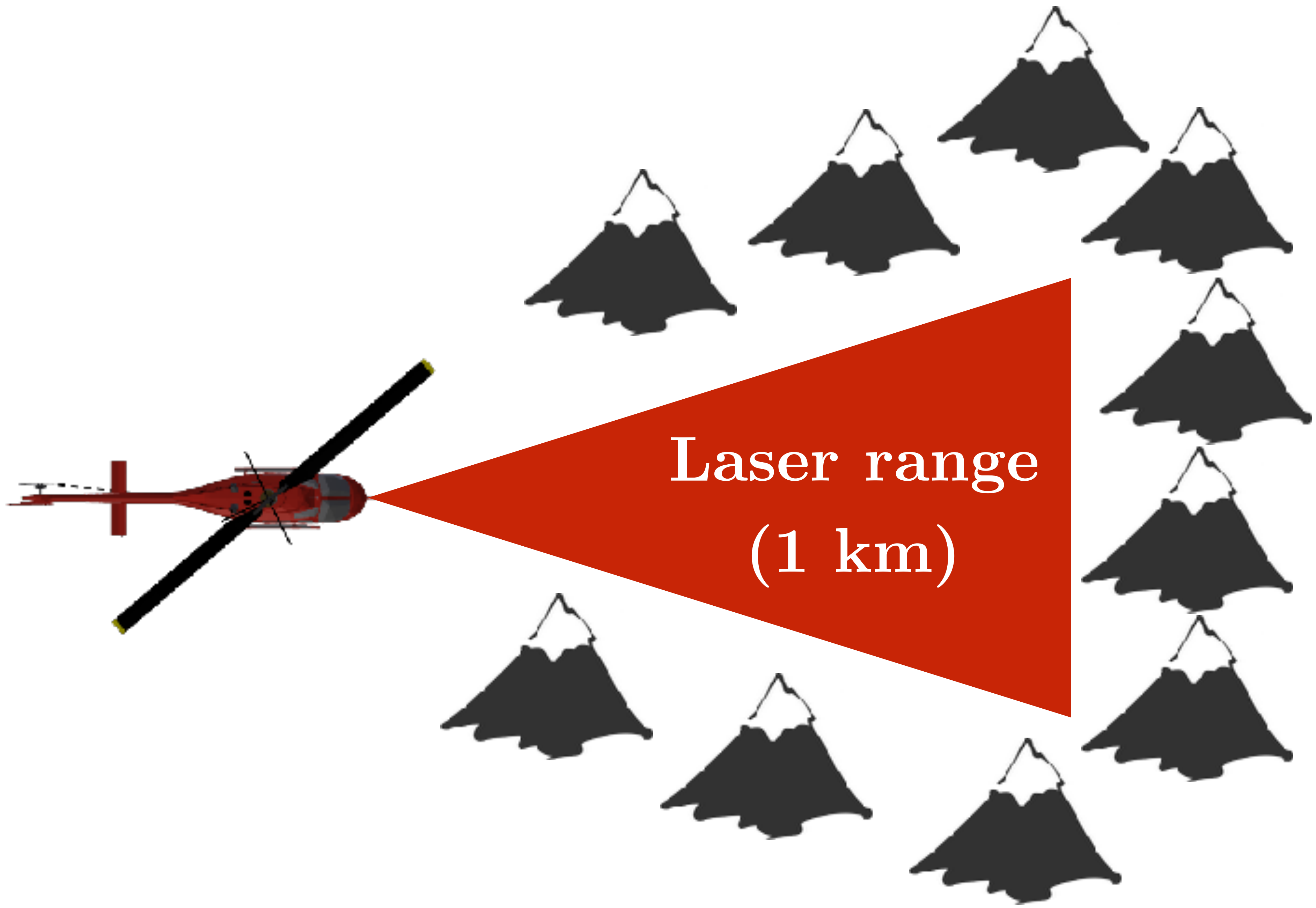




G. Hollinger, B. Englot, F. Hover, U. Mitra, and G. Sukhatme, "Active planning for underwater inspection and the benefit of adaptivity," International Journal of Robotics Research (IJRR), vol. 32, no. 1, pp. 3-18, Jan. 2013.

Safety

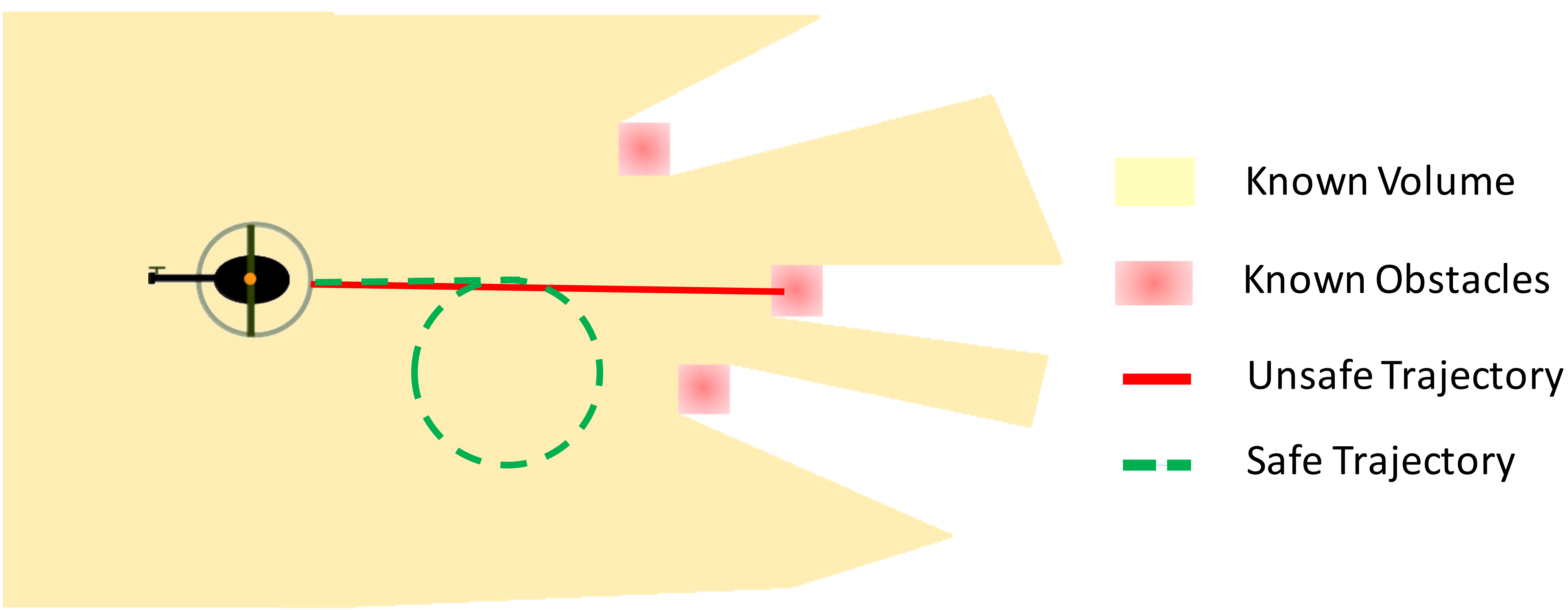
Guaranteeing safety



What prevents the system from flying at high speeds to a dead end?

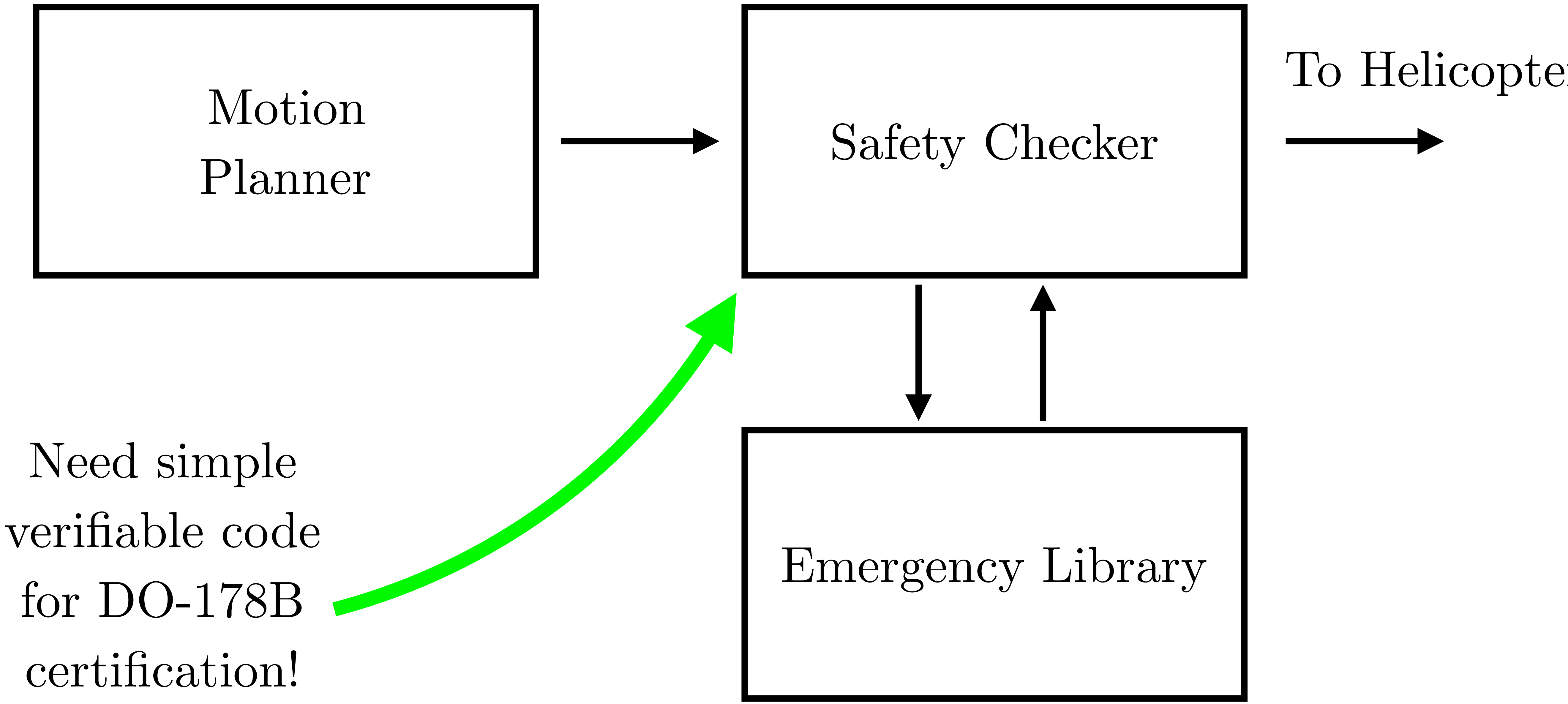
Safety planner that guarantees the robot can stay safe

What is a safe state?



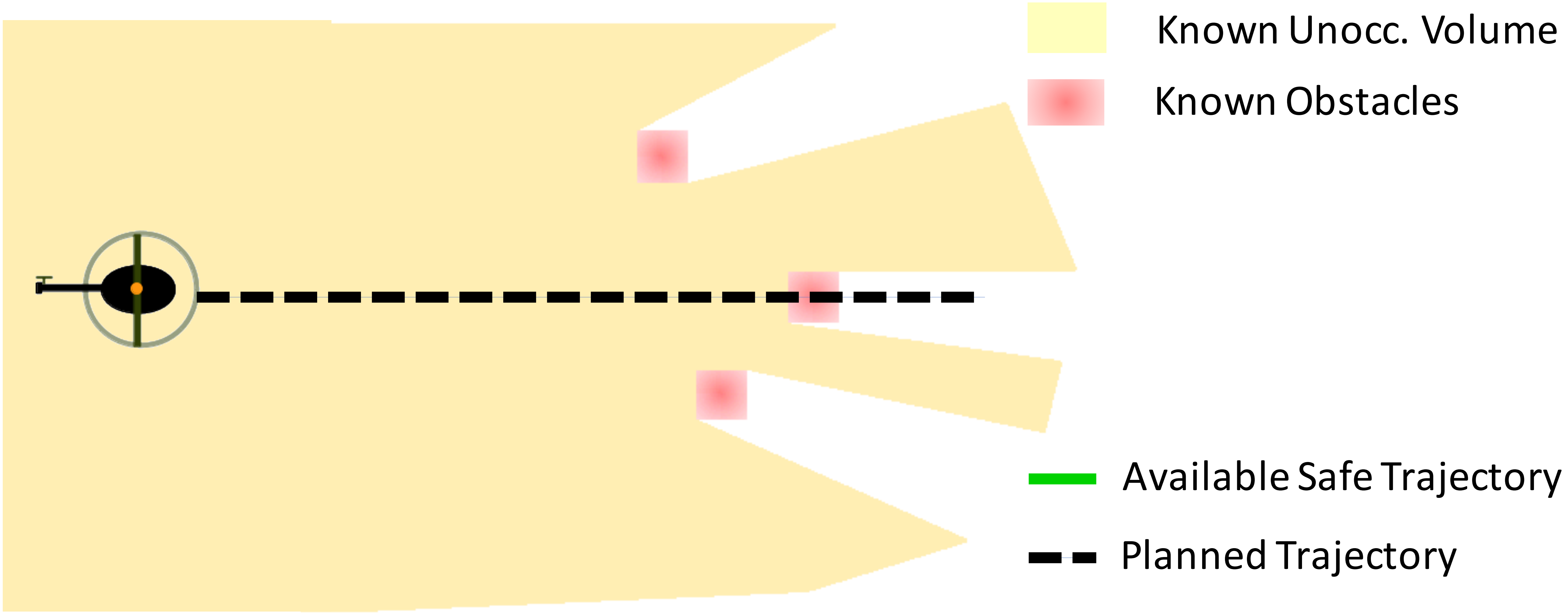
Must exist a trajectory in known free space

Guaranteed Safe Planning

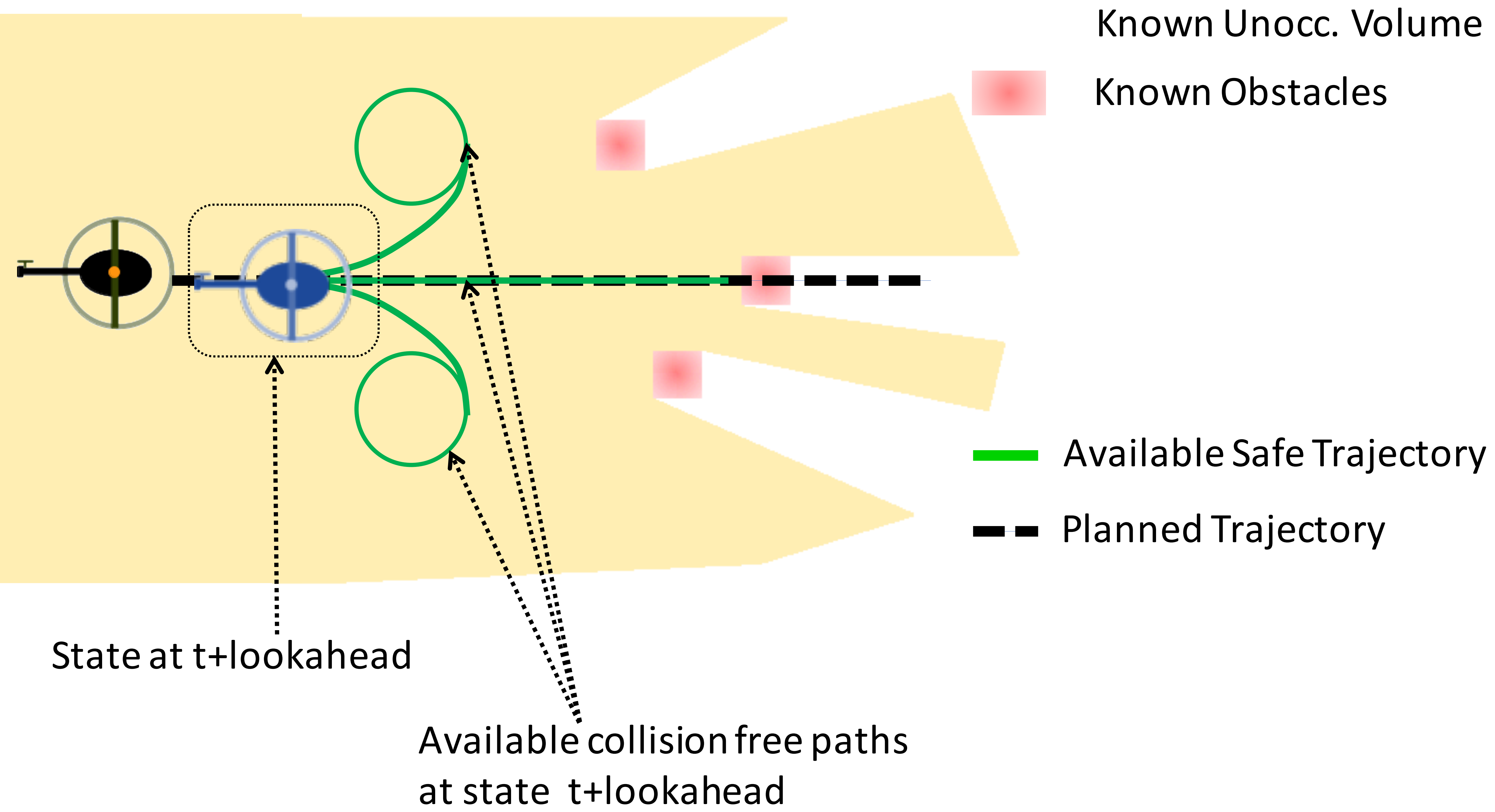


1. S.Arora, **S.Choudhury**, D. Althoff and S. Scherer. "Emergency Maneuver Library – Ensuring Safe Navigation in Partially Known Environments", ICRA (2015)

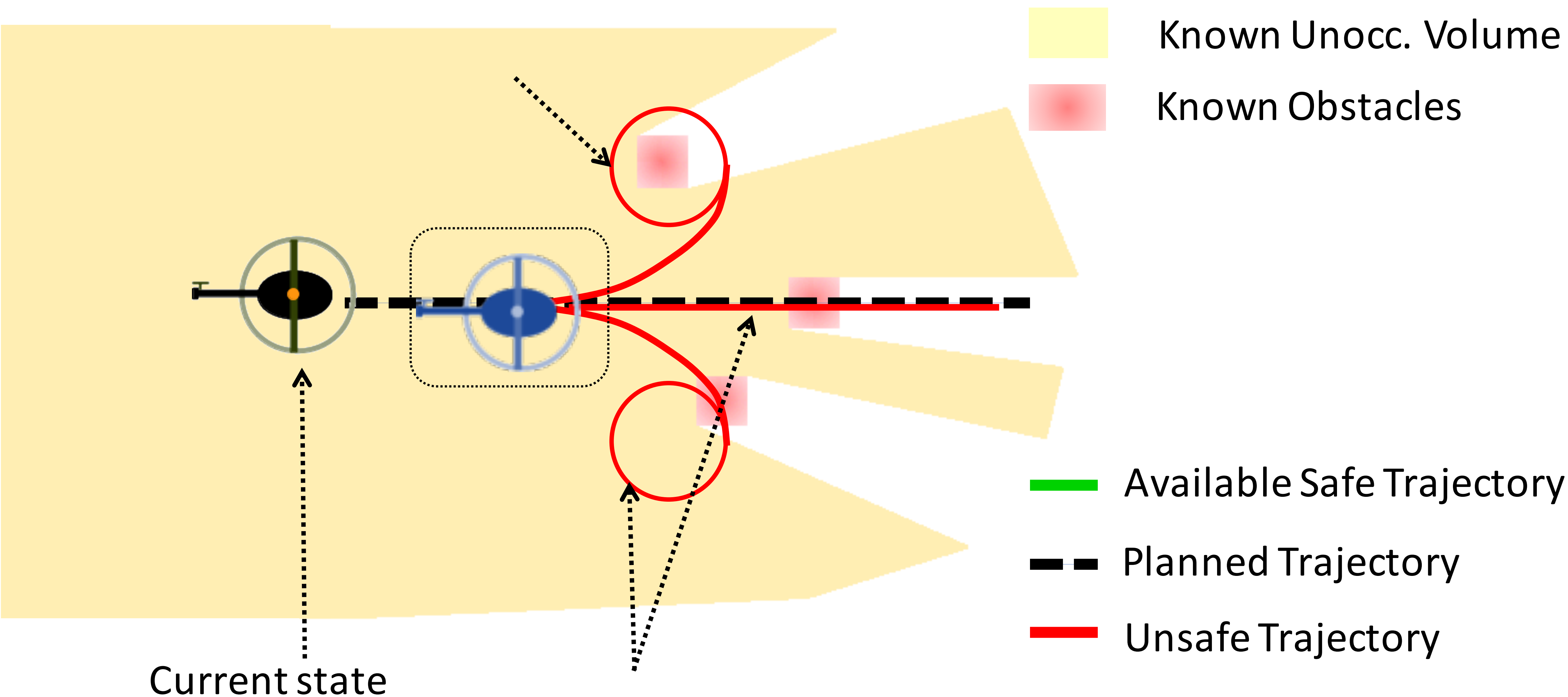
Safety Algorithm



Safety Algorithm

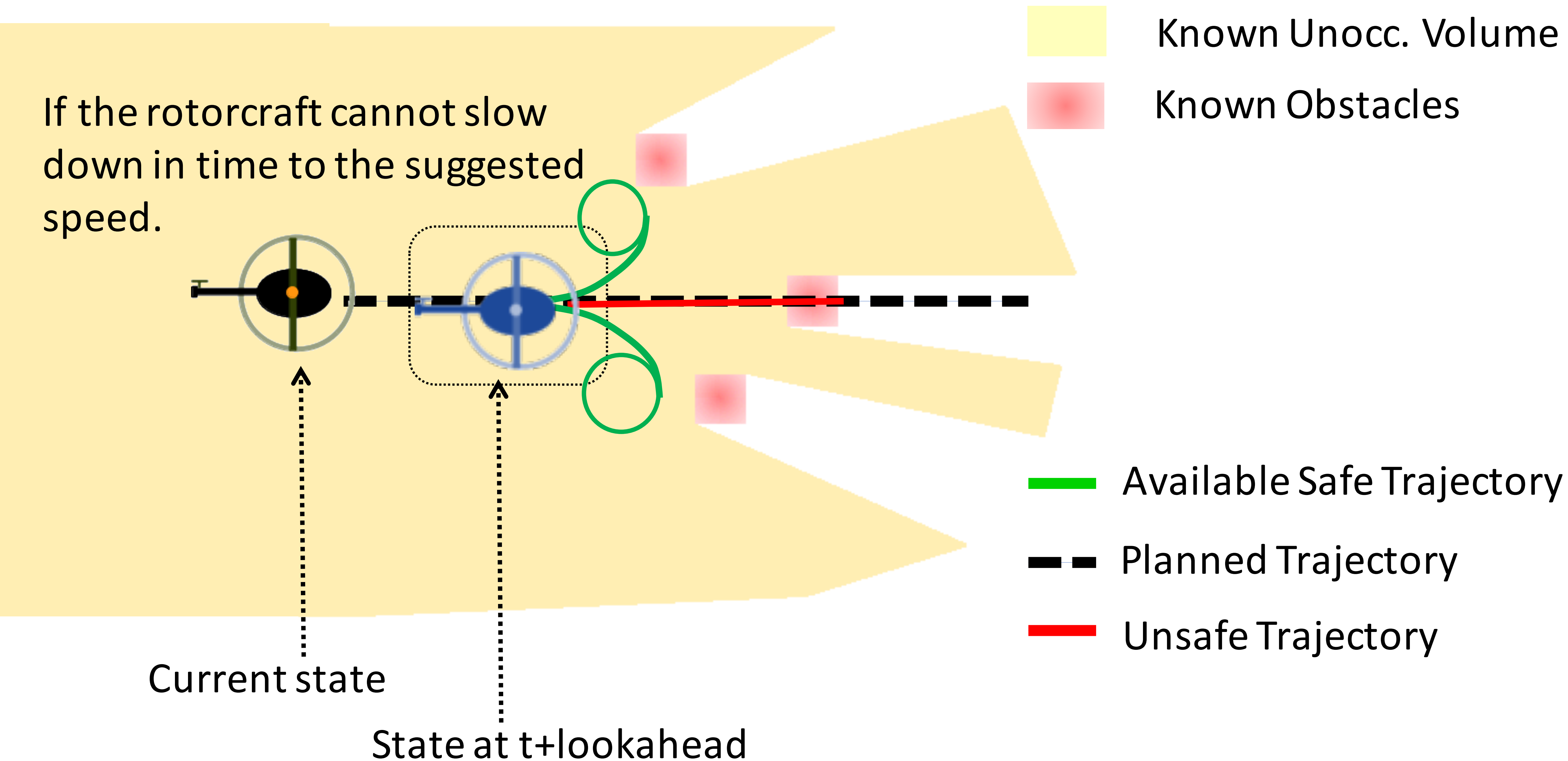


Safety Algorithm

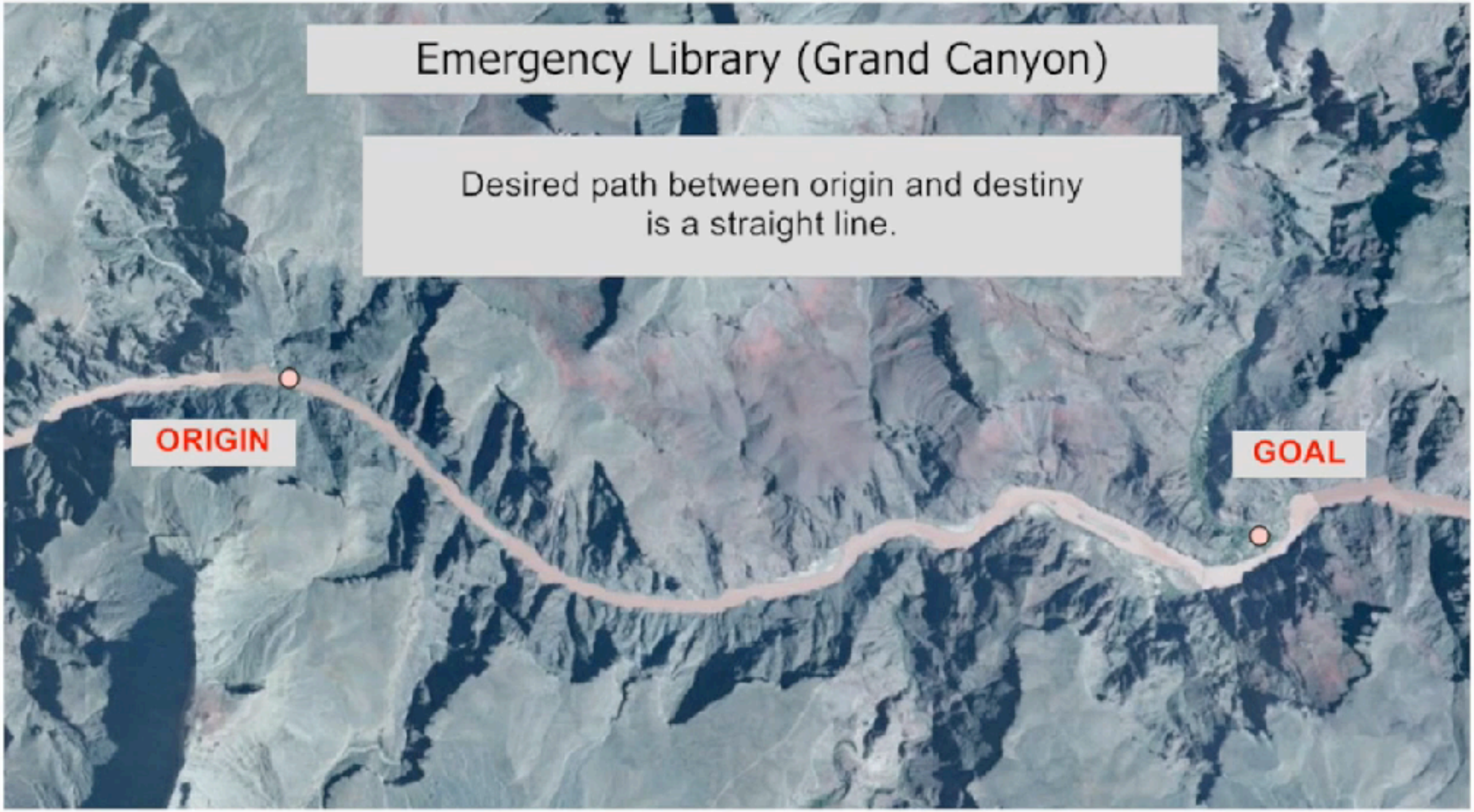


Safety Algorithm

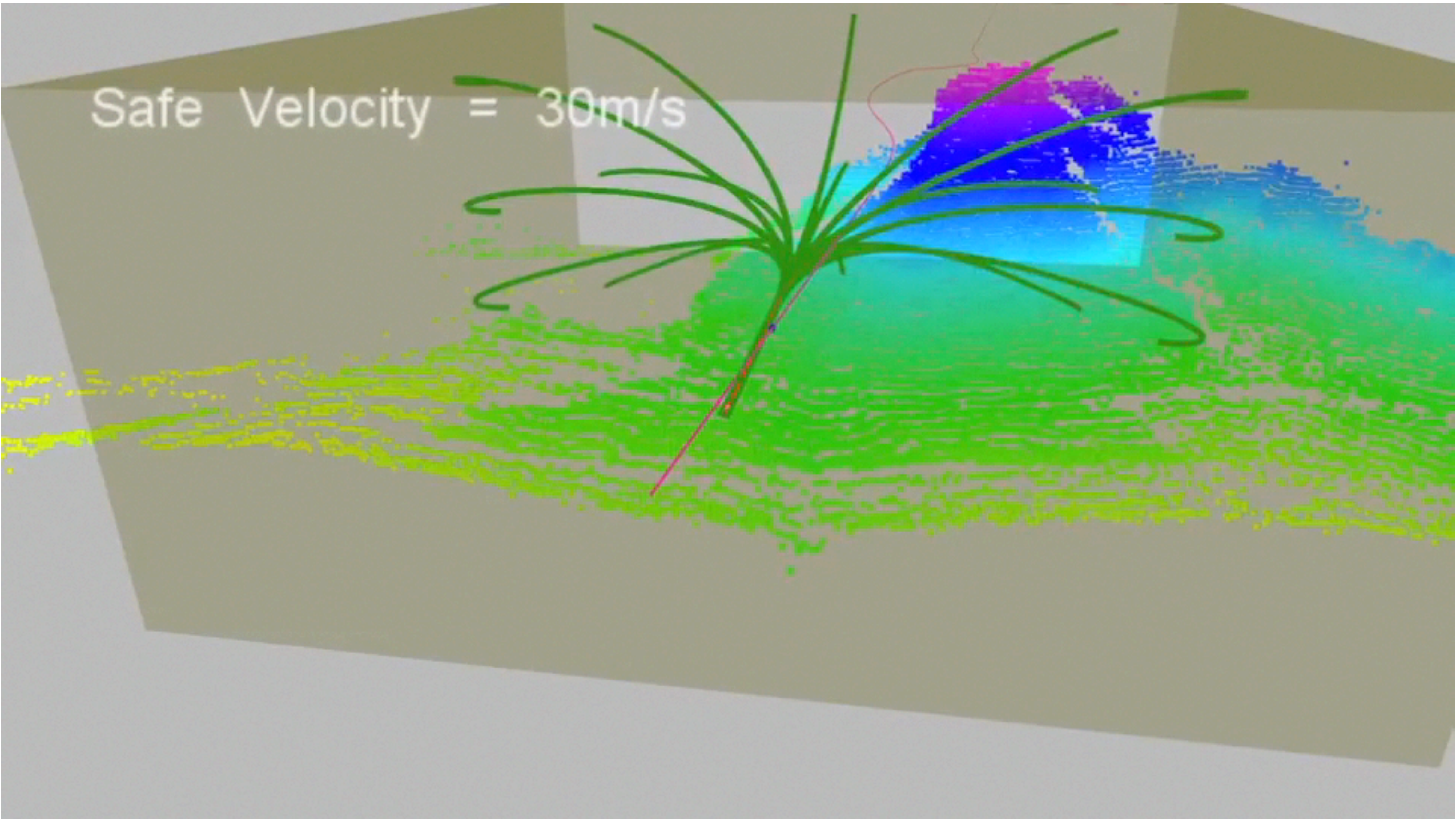
If the rotorcraft cannot slow down in time to the suggested speed.



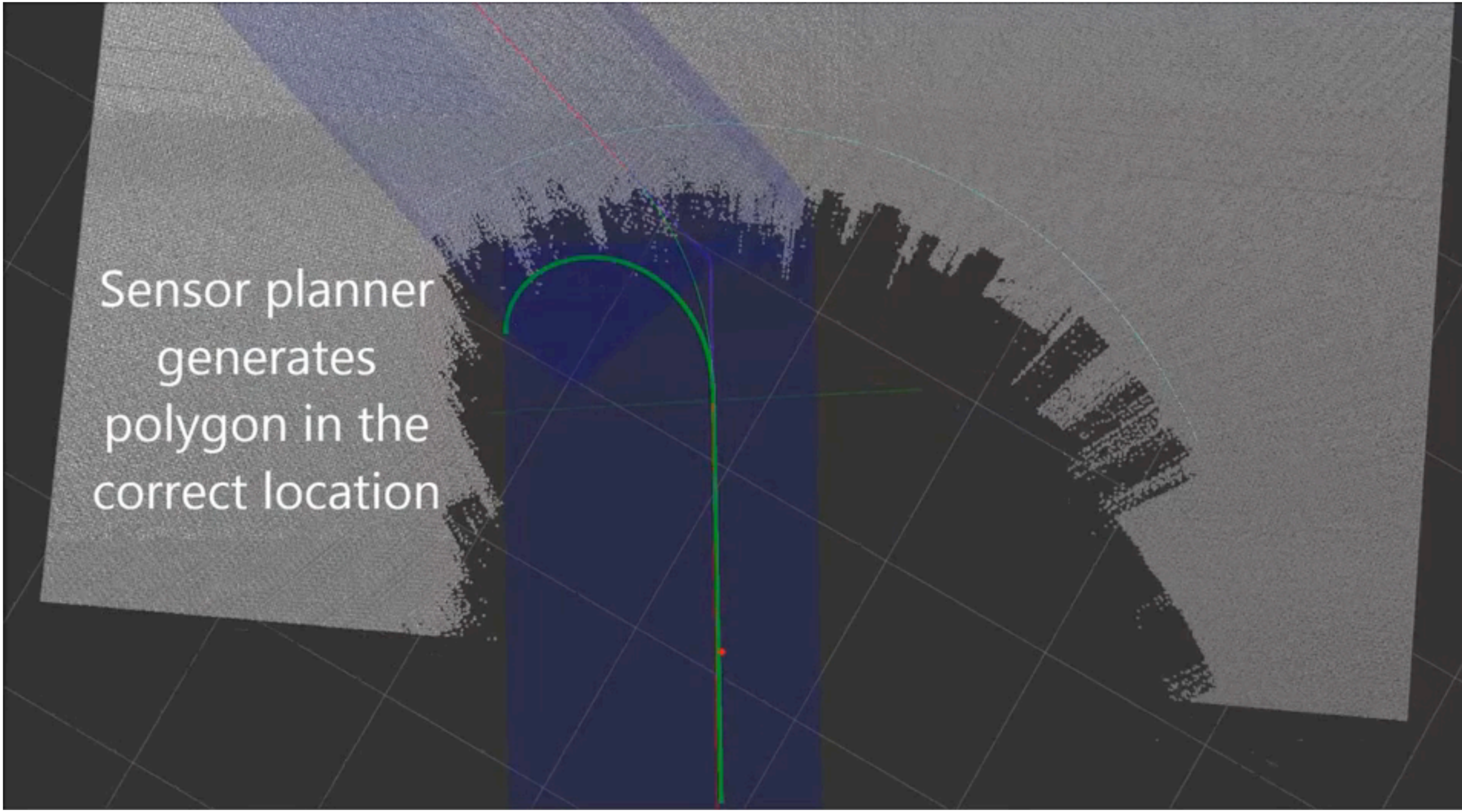
When is a safety maneuver triggered?



Guaranteeing safety at close obstacle proximity



Safety during actual sensor failure



How do we compute a library of emergency maneuvers?

Generating the Emergency Maneuver Library

$$\Phi_d = \arg\max P_u(\Phi)$$

Subject to: $\Phi_d \leq N_d$

Where,

$P_u(\Phi)$ is the probability of a trajectory being unobstructed in the trajectory Φ

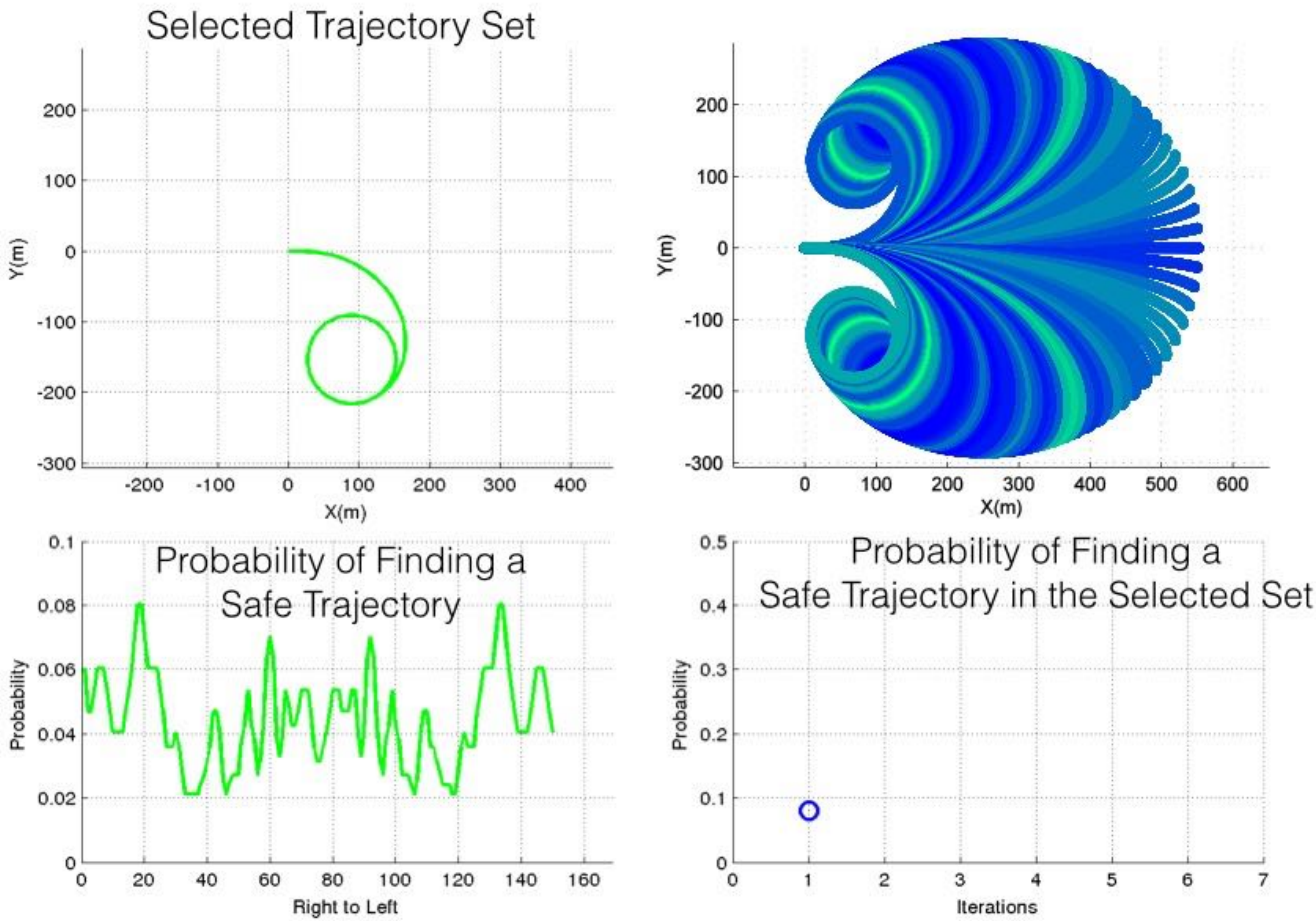
N_d is the number of trajectories that can be checked for obstruction in real time

NP-hard problem.

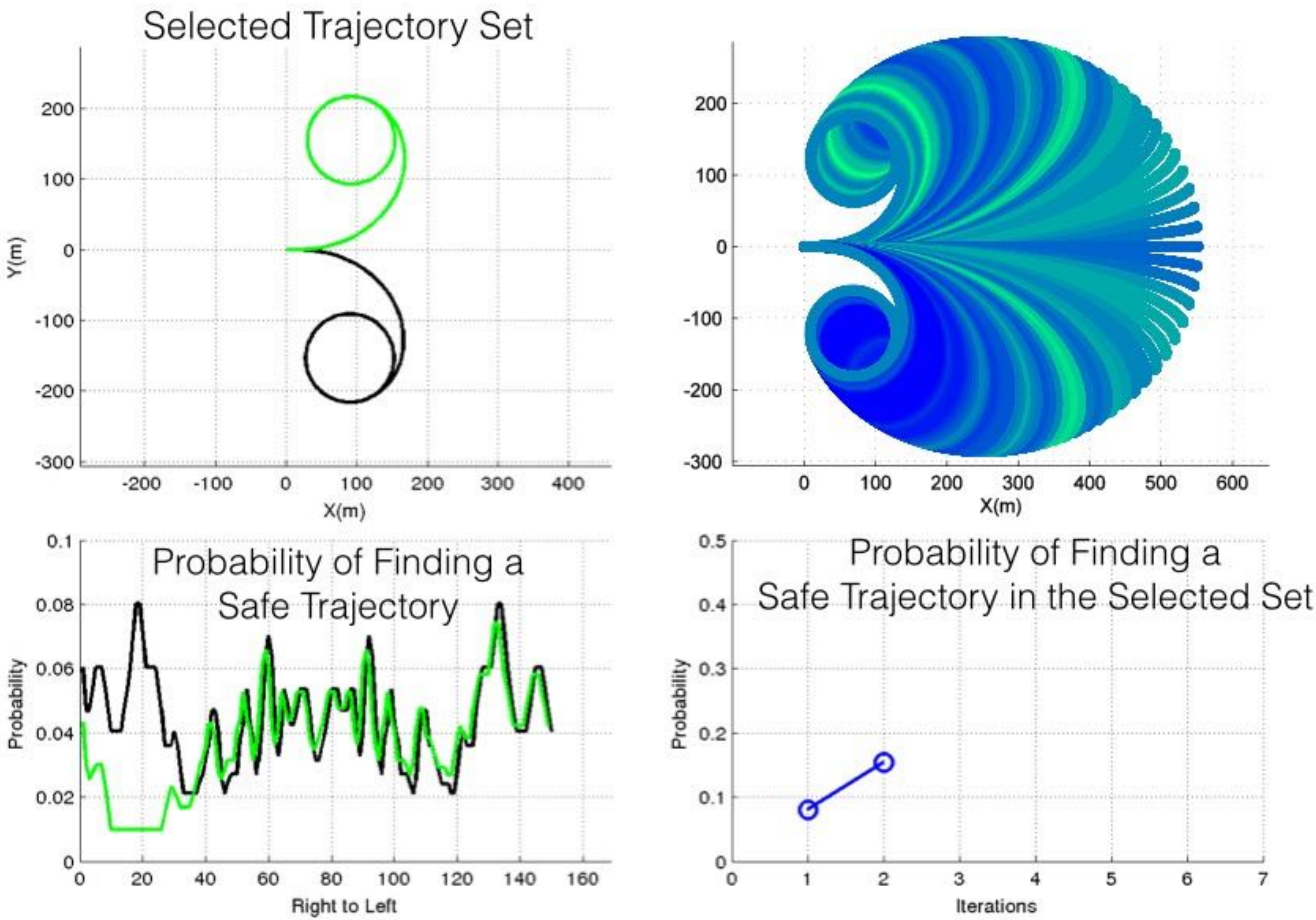
Sub-modular, monotonic structure leads to an efficient greedy algorithm which allows for near-optimal solutions

Arora et. Al, "Emergency Maneuver Library – ensuring safe navigation in partially known environments", ICRA 2015.

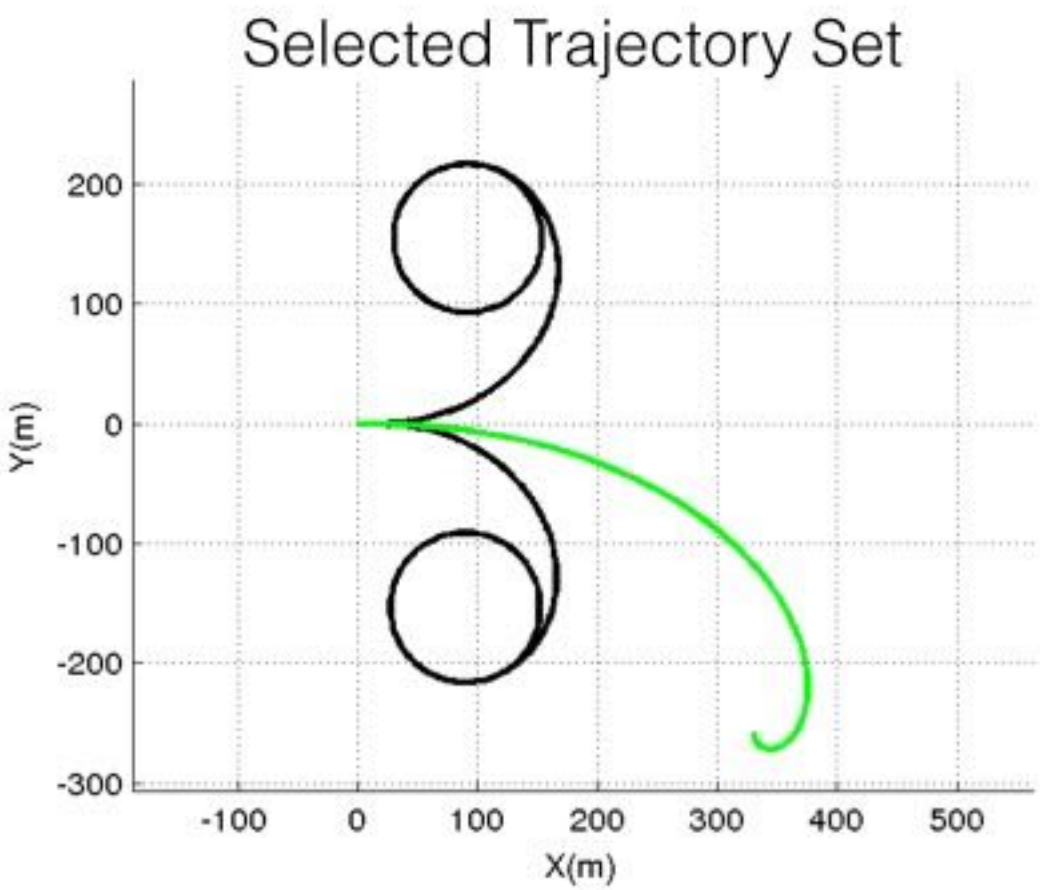
Library Construction

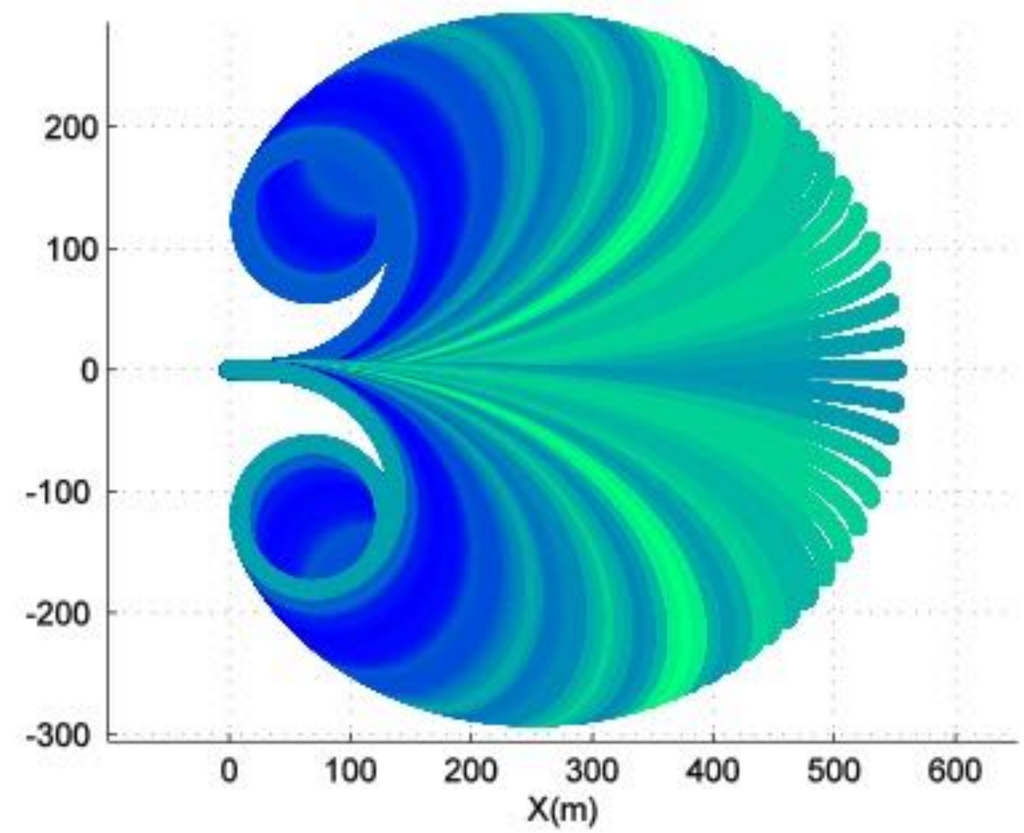


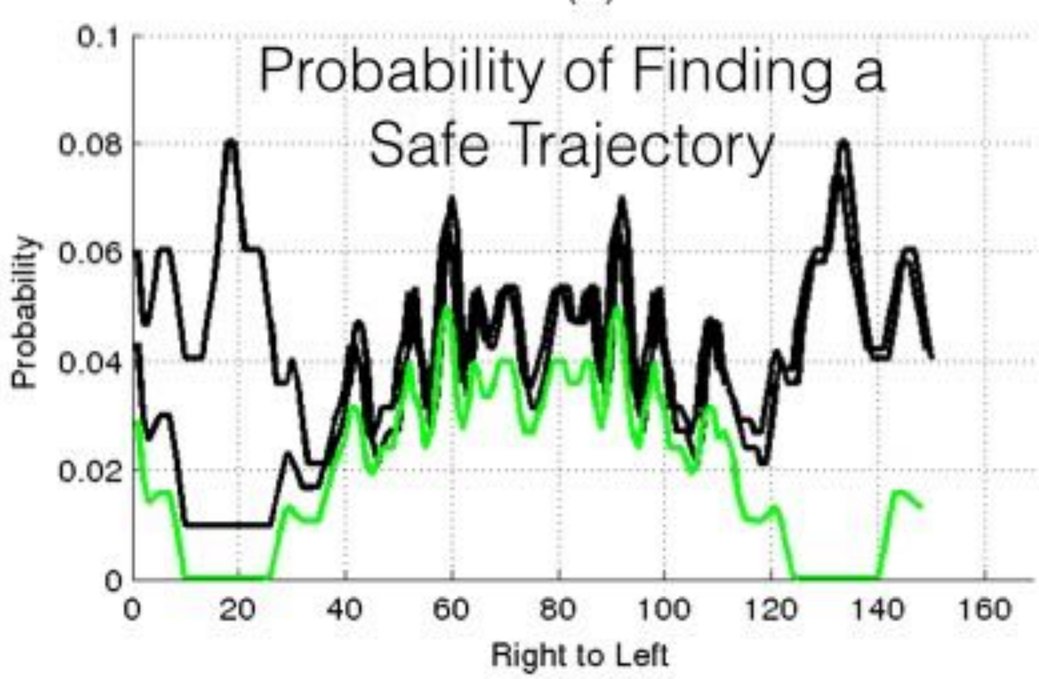
Library Construction

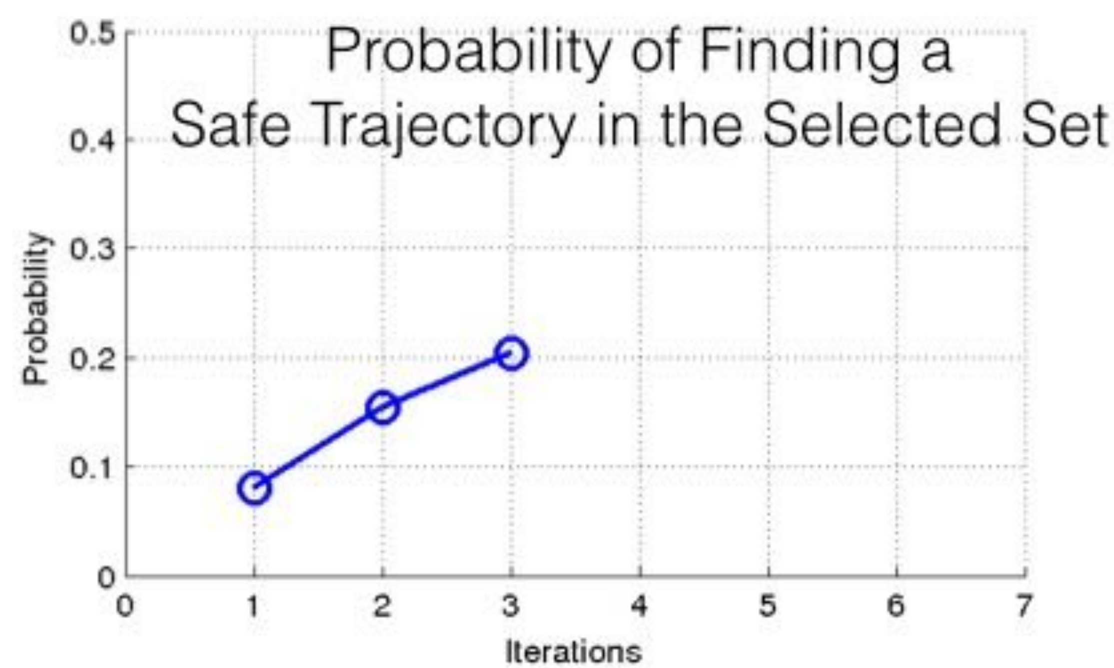


Library Construction

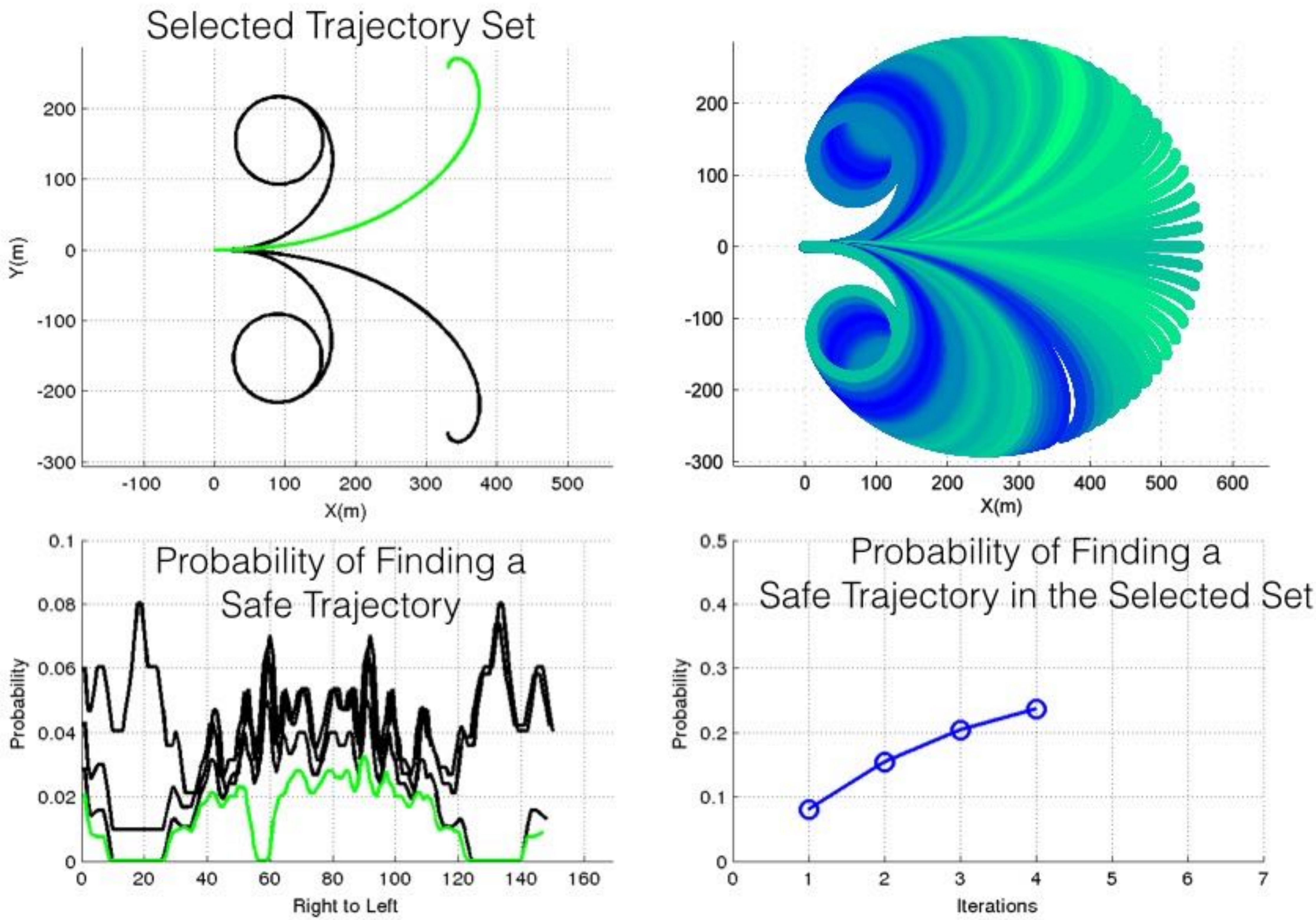




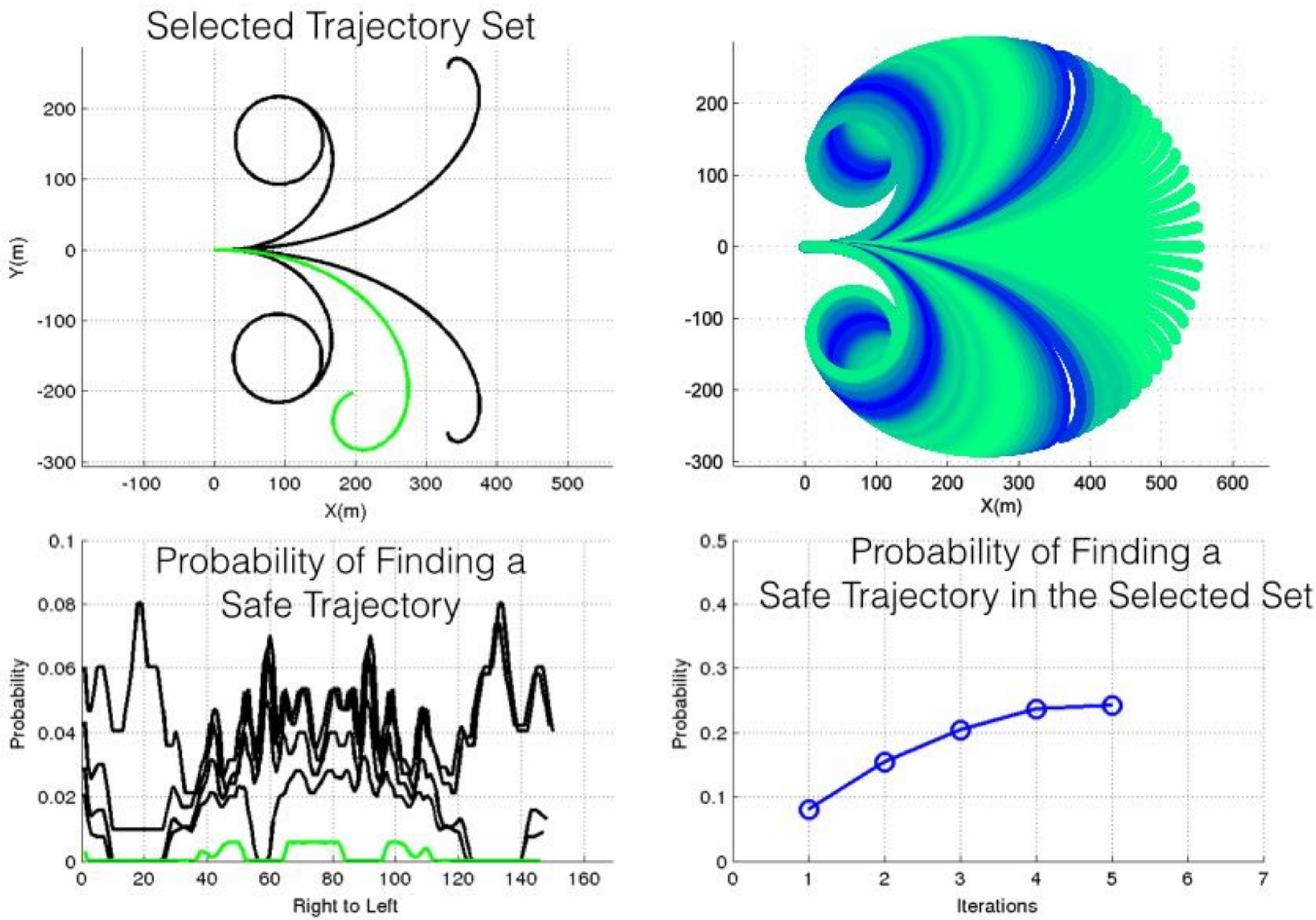




Library Construction



Library Construction



Library Construction