air lab

Motion Planning for Flying Robots

Sebastian Scherer

http://theairlab.org

at the Field Robotics Center

Carnegie Mellon University

Outline

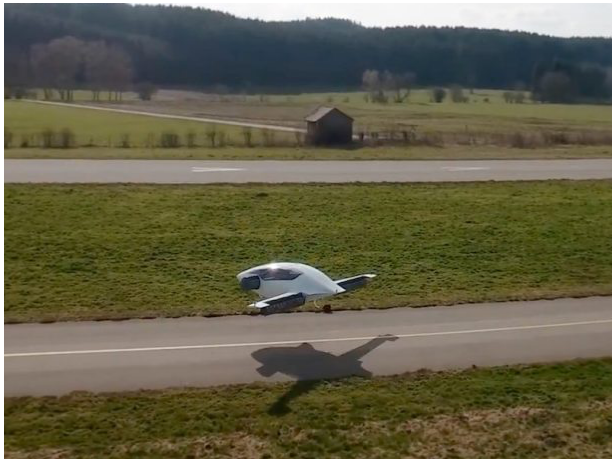
- **Introduction**
- Problem
- Abstraction and Approach
- Results
- Summary
- Exercise

Once a dream…

Uber Elevate Vertiports

Lilium Operation Concept

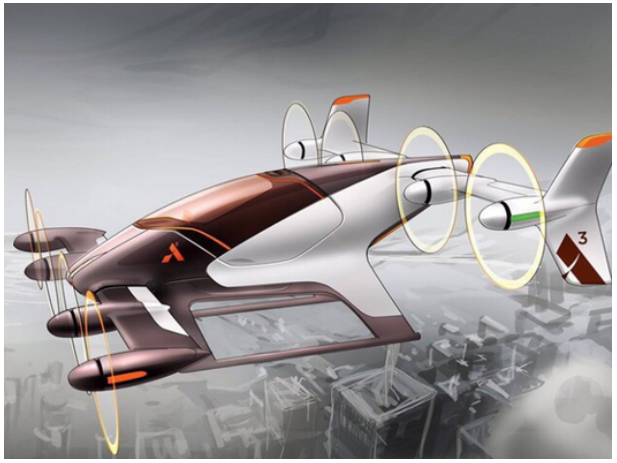
… becoming a reality



Lilium

eHang

Aurora



Airbus A^3

What has changed?

- e-VTOL have the potential to be much lower operating cost due to the simplicity of electric designs and active control.

- If the operating cost is low enough urban air taxis may make economic sense.

Why is autonomy relevant for these systems?

- Current vertical flight safety expectation don't match passenger expectations.

- Not enough pilots for the scale of operation and take a long time to train.

Total number of eligible pilots in the US: 25,918
(FAA US Civil Airmen Statistics 2016)

- Congested and concerted air traffic can be beyond pilots capability.

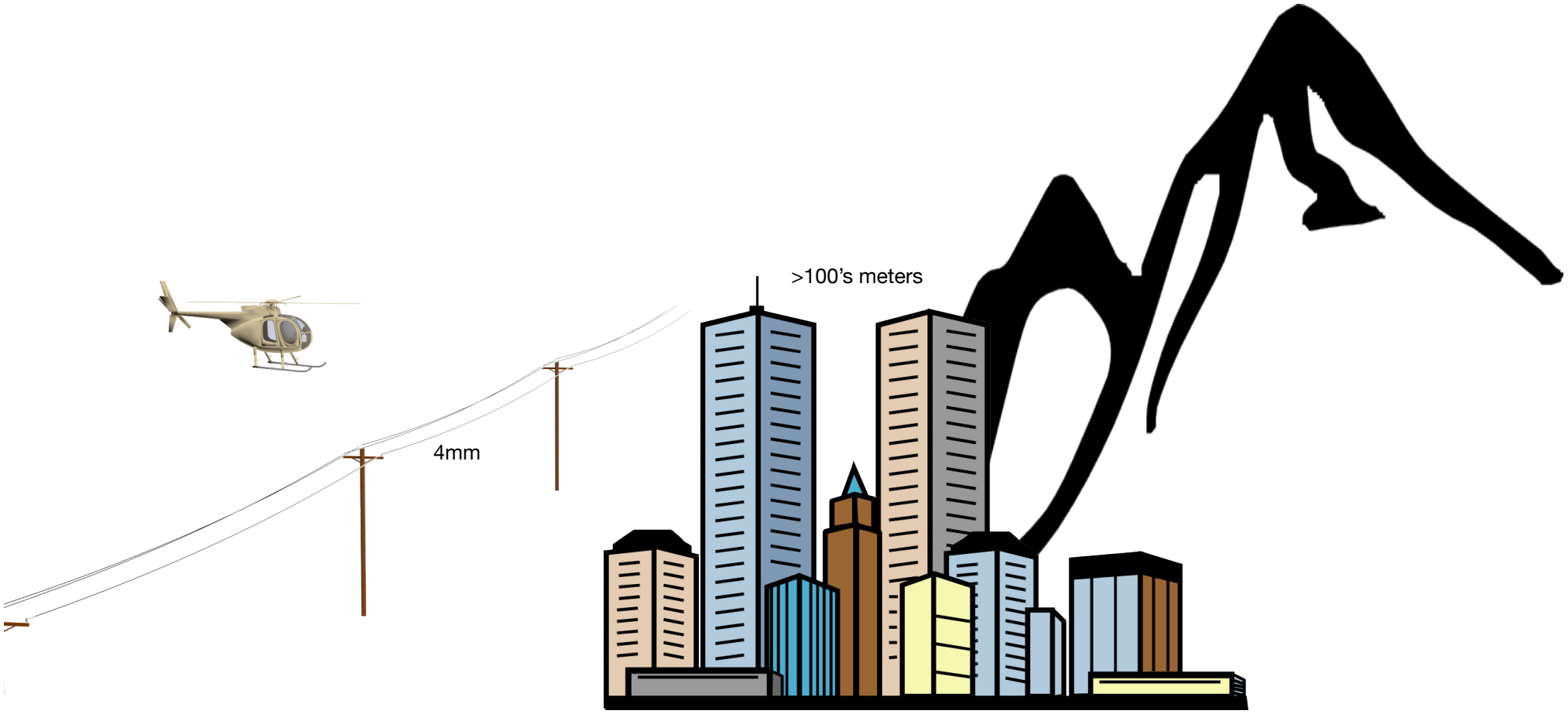
- 200 pounds of extra batteries make a big difference ~ another 20kWh

Assuming 200 Wh/kg

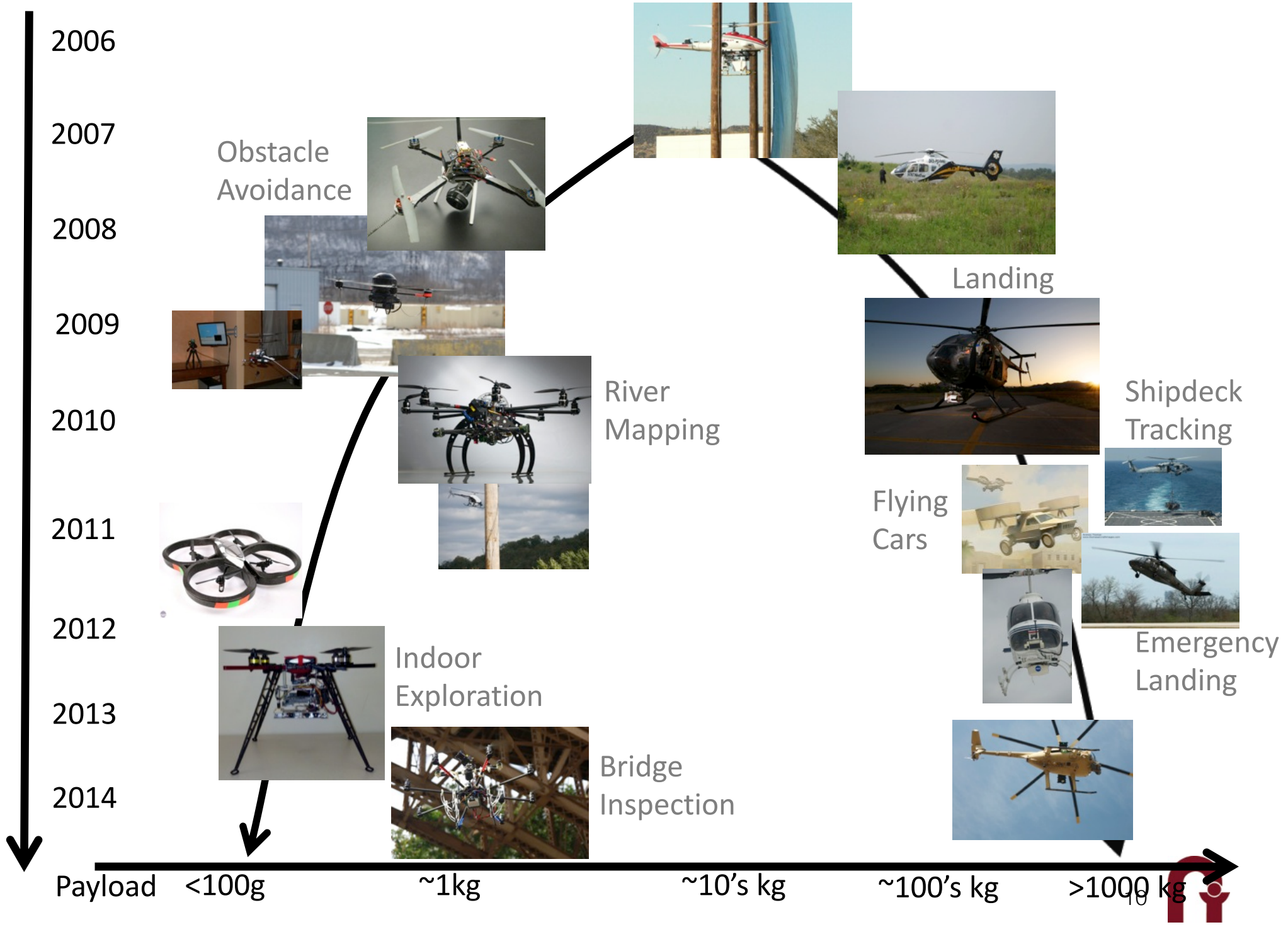
WSJ 2009, https://www.wsj.com/articles/SB123370317612745375

Why is Autonomy Difficult?

Why is Autonomy Difficult?



Why is Autonomy Difficult?


2006
2007
2008
2009
2010
2011
2012
2013
2014
Obstacle Avoidance
River Mapping
Indoor Exploration
Bridge Inspection
Landing
Shipdeck Tracking
Flying Cars
Emergency Landing
Payload
<100g
~1kg
~10's kg
~100's kg
>1000 kg

Small Example

Large: Autonomous Approach and Landing

Choudhury et al. AHS Forum 2014, and other work

Each System Operates in Different Environments and Has a Different Motion Planning Approach

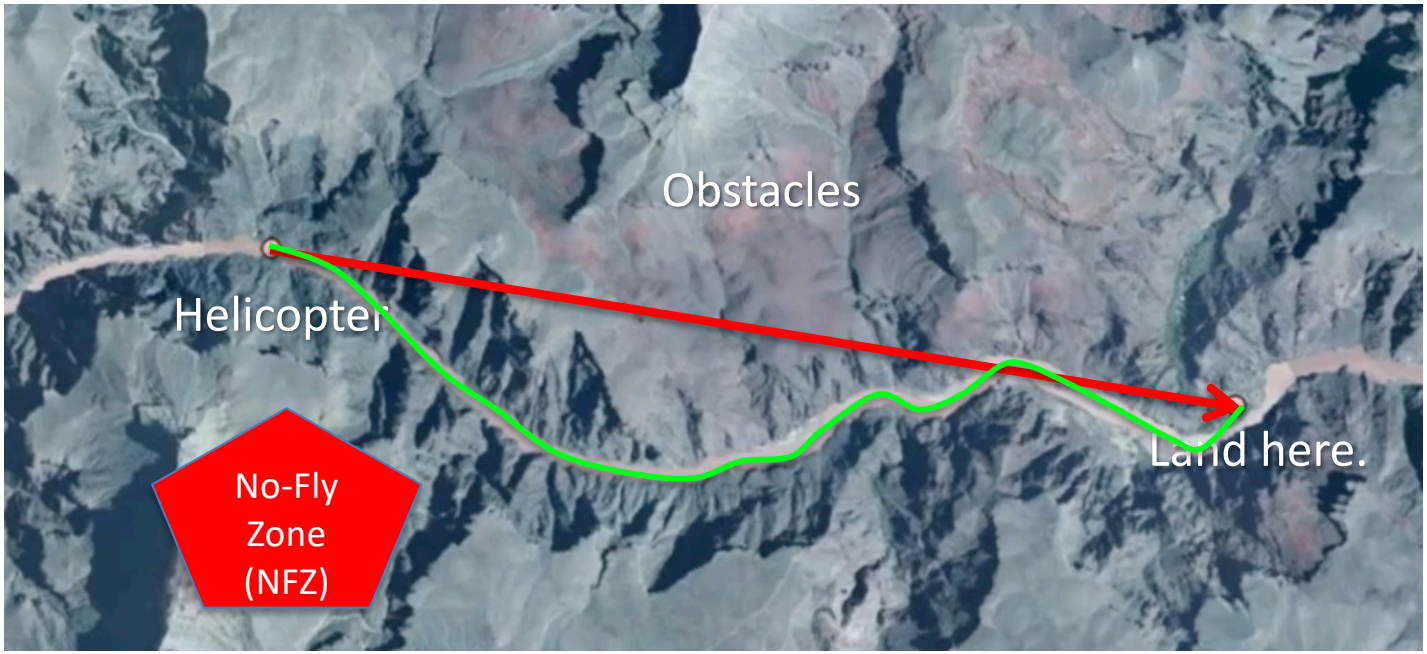
- Why don't we have an ultimate motion planning system that works well for all applications?
- What is common and what is different between applications?
- What are potential approaches?

Outline

- Introduction
- **Problem**
- Abstraction and Approach
- Results
- Summary
- Exercise

Motion Planning Problem

React in real-time to previously unknown obstacles, avoid no-fly-zones, and land.



Assumptions

Here:
- Little uncertainty
- No exploration actions necessary

Variations on the problem

System uncertainties:
- Position
- Sensing
- Action

Need to gather data about the world:
- Maximize information gain
- No explicit goal state
- Viewpoint planning or active exploration

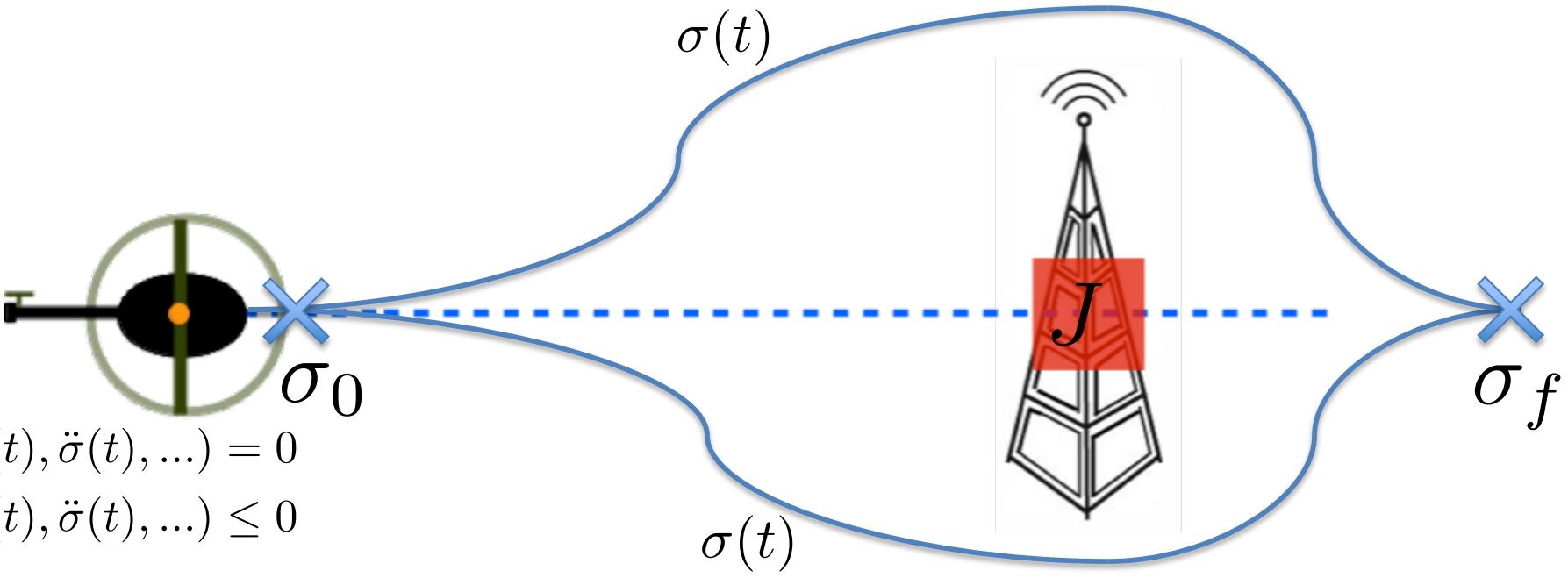
The Trajectory Planning Problem

find	$\sigma(t) = \{x(t), y(t), z(t), \psi(t), t_f\}$	Time parameterized trajectory
minimize :	$J = \int_0^{t_f} c(\sigma(t))dt + c(\sigma(t_f))$	Cost function
constraints :	$\sigma(0) = \sigma_0$ $\sigma(t_f) = \sigma_f$	Boundary value constraints
	$h(\sigma(t), \dot{\sigma}(t), \ddot{\sigma}(t), \ldots) = 0$	Non-holonomic constraints
	$g(\sigma(t), \dot{\sigma}(t), \ddot{\sigma}(t), \ldots) \leq 0$	System limitations
	$J < \infty$	Cost function constraints

Variant of the optimal control problem constrained to a trajectory

Note that typically *J* is partially known and discovered on the fly.

Example



$$h(\sigma(t), \dot{\sigma}(t), \ddot{\sigma}(t), ...) = 0$$
$$g(\sigma(t), \dot{\sigma}(t), \ddot{\sigma}(t), ...) \leq 0$$

Example non-holonomic constraint:

$$h(\sigma(t), \dot{\sigma}(t)) = w\dot{q} = 0$$

$$\dot{q} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} \quad w = \begin{pmatrix} \sin\theta & -\cos\theta & 0 \end{pmatrix}$$

Example system constraint:

$$g(\sigma(t), \dot{\sigma}(t)) = \|\dot{\theta}_{max} - \dot{\theta}\| \leq 0$$

Planning Problem: Cost Functions

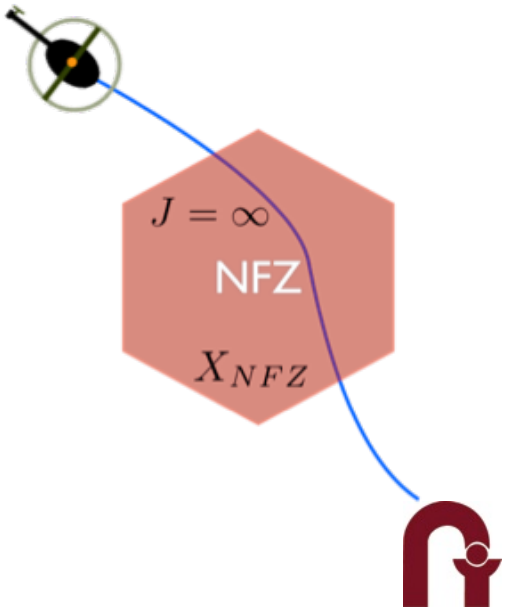
$$J_{total} = \sum_i w_i J_i$$

1. Time to Mission Completion

$$J_1 = t_f$$

2. No Fly Zone

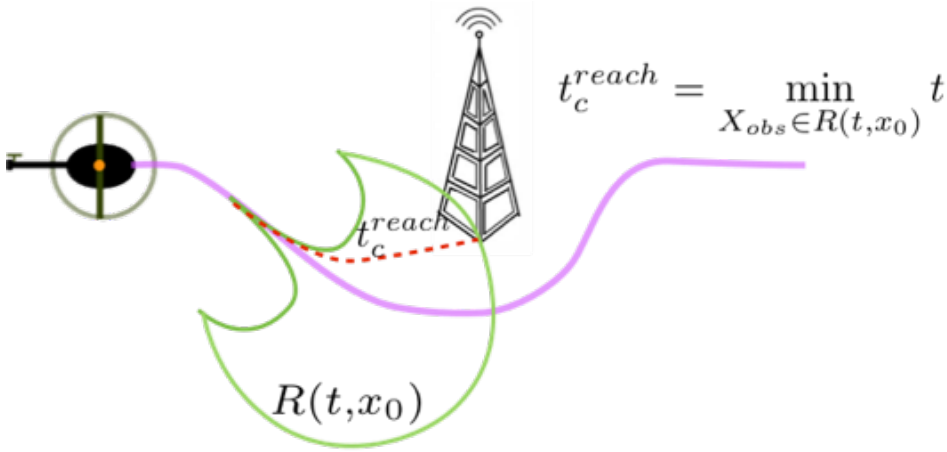
$$J_2 = \begin{cases} \infty & \sigma(t) \in X_{NFZ} \\ 0 & otherwise \end{cases}$$



Planning Problem: Cost Functions

3. Time to Collision

- Representation of collision risk
- Scales with velocity
- Based on reachability volume
- Approximated by a pyramid



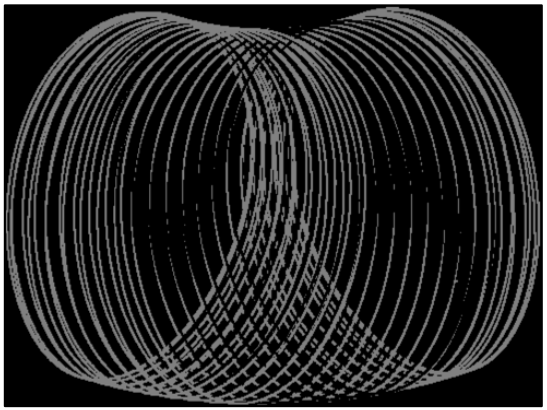
$$t_c^{reach} = \min_{X_{obs} \in R(t,x_0)} t$$

(Reachability)

$$J_3 = \begin{cases} (t_{max} - t_c^{reach})^2 \\ \infty \end{cases} \qquad d_{obs} < d_{critical}$$

Obstacle Representation

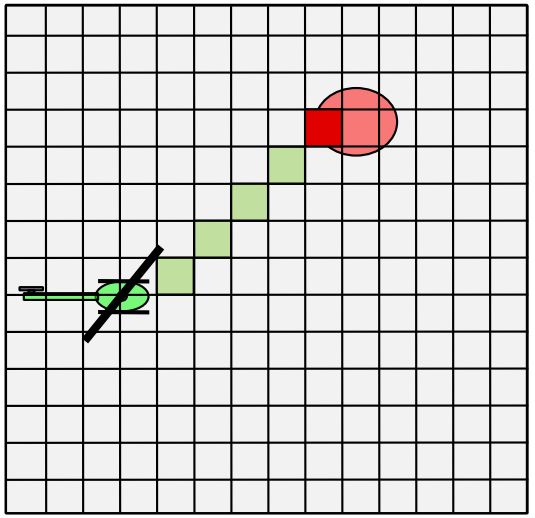
Obstacle



Range vector
measurement

Sensor
Model

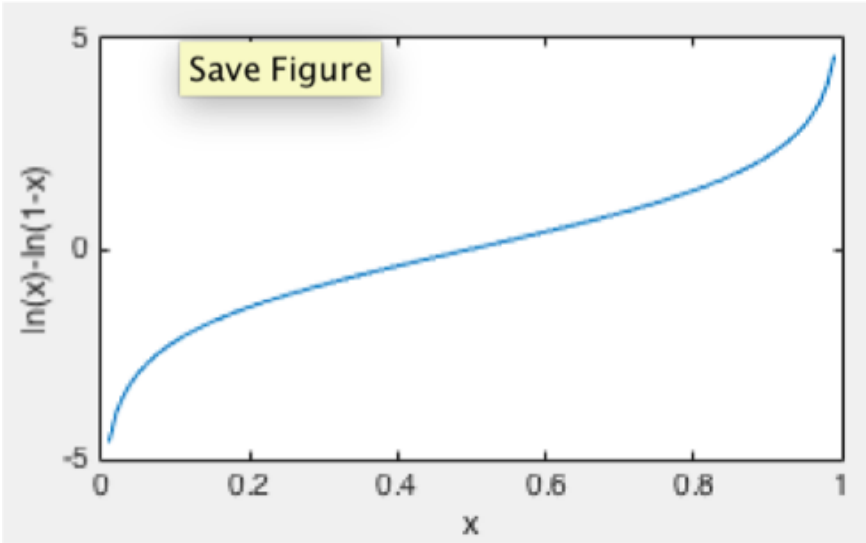
P(o | m)



3D Evidence Grid

Updating an Evidence Grid Cell

- Each cell is assumed to be independent and contains the belief of occupancy of the volume
- The belief can be updated as follows (assuming the prior *P(m)=0.5)*:



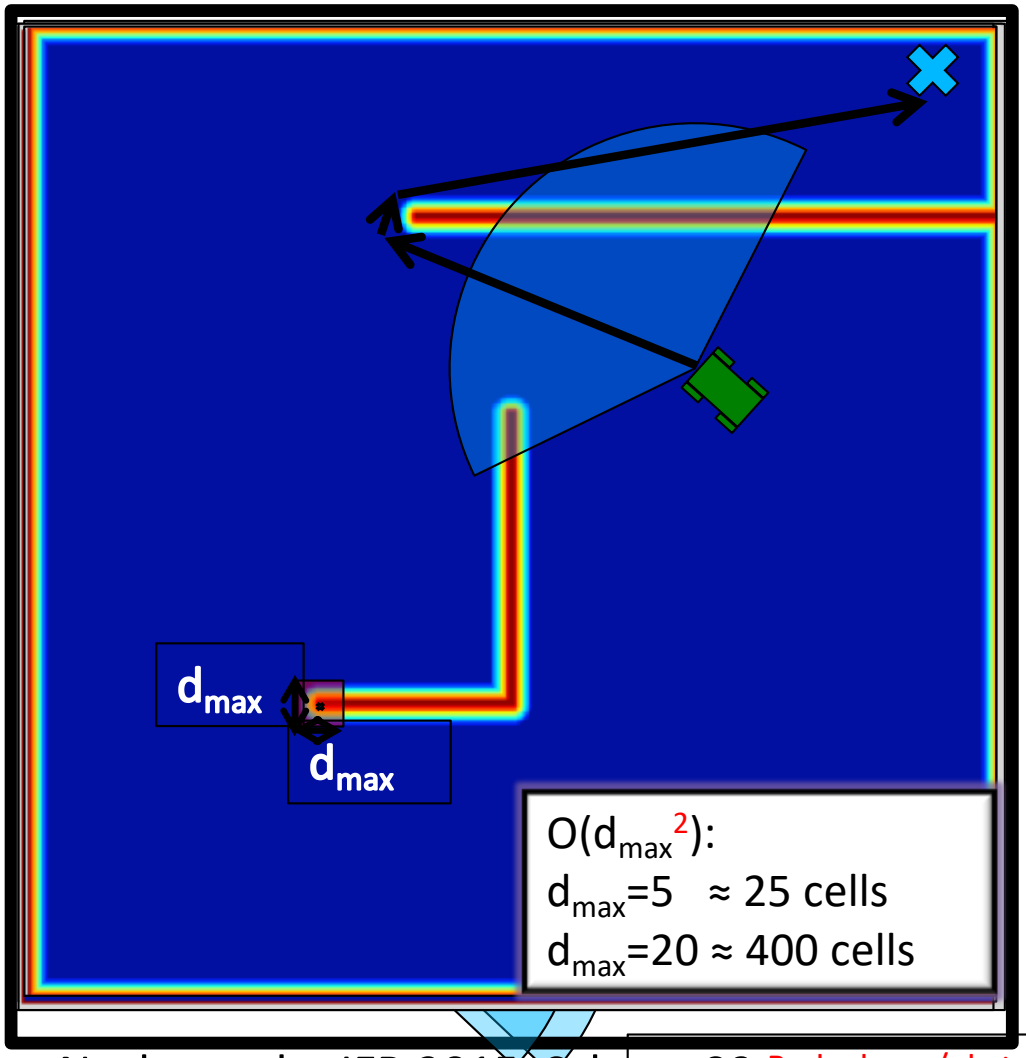
$$b'(m) = 1 - \left(1 + \frac{P(m|o)}{1-P(m|o)} \cdot \frac{b(m)}{1-b(m)}\right)^{-1}$$

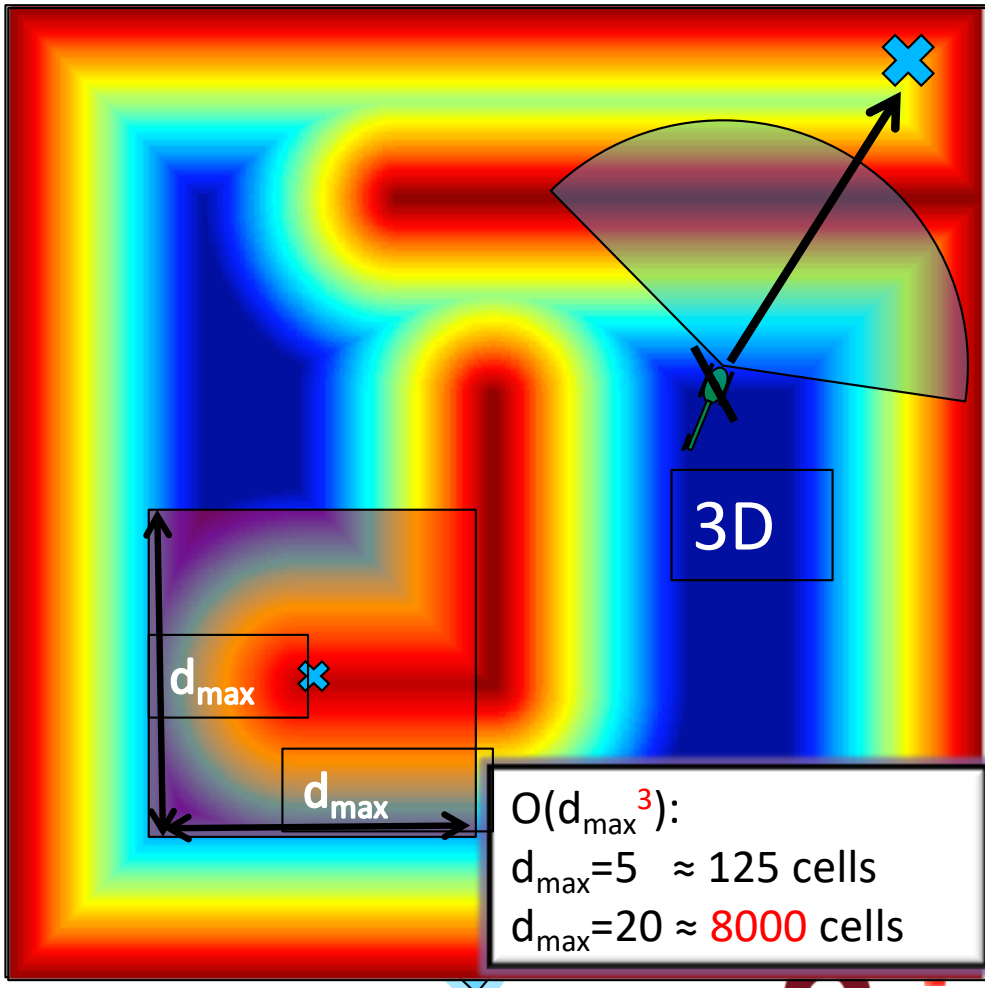
- However usually a log-odds representation is used:

$$\bar{b}(m) = \ln \frac{b(m)}{1-b(m)} \qquad \bar{b}'(m) = \bar{b}(m) + \ln P(m \mid o) - \ln\left(1 - P(m \mid o)\right)$$

Foessel02, Moravec

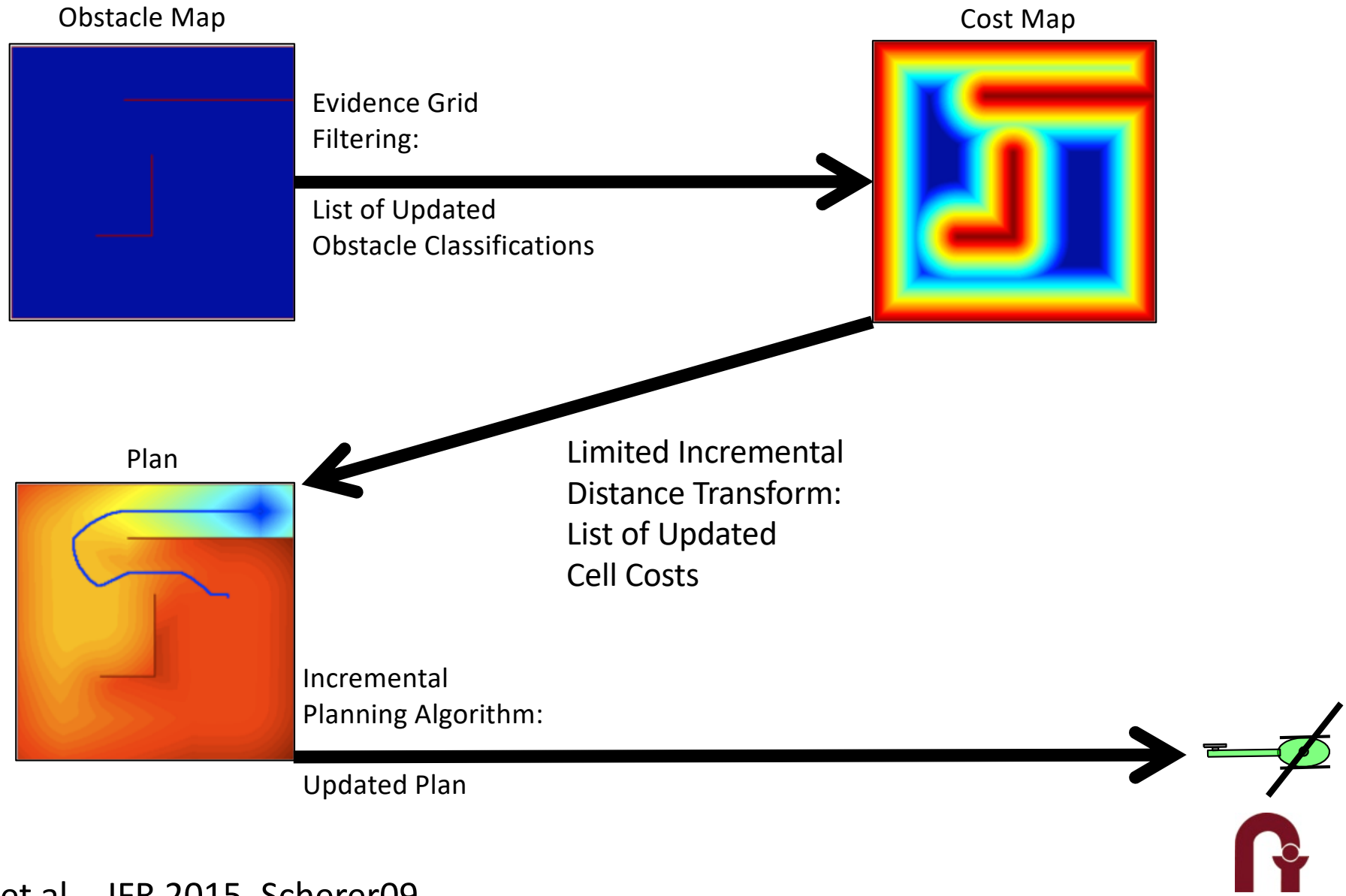
A typical Sense, Plan, and Act Cycle (Comparison between Ground and Air Robots)





Nuske et al. , JFR 2015, Scherer09 Red = large/obstacle Blue = small/free

A Completely Incremental Framework for Planning



Nuske et al. , JFR 2015, Scherer09

How does one handle large environments and speeds?

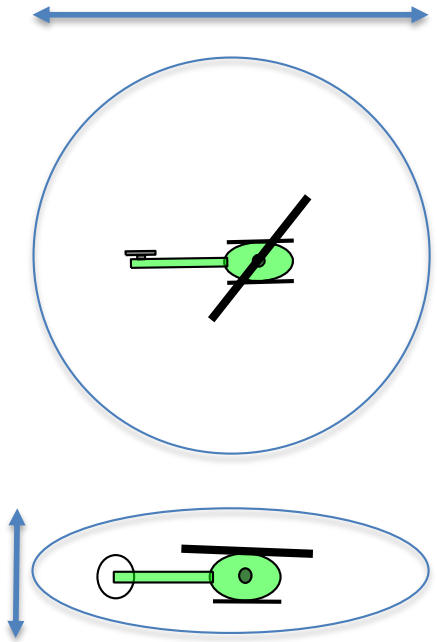
- Full-scale helicopter at 60 m/s. Mission lengths ~ 400km.
- Scrolling buffer grids:

$$i = rx \mod n$$

$$j = ry \mod n$$

$$k = r_z z \mod m$$

- r, r_z = resolution, n,m = map size,
- x,y,z = coordinates, l,j,k = map indices



Nuske et al. , JFR 2015

Questions

- What are some of the pros/cons of this type of approach for filtering?
- What are alternative approaches to represent the world what are their advantages and disadvantages?

Planning Problem: Constraints

1. Dynamics

Coordinated vehicle motion with no sideslip (above 20 knots)

$$\psi = \tan^{-1}\frac{\dot{y}}{\dot{x}}$$

$$\phi = \tan^{-1}\frac{v\dot{\psi}}{g}$$

ψ heading

ϕ roll

2. Actuator Limits

$$\phi \leq \phi_{max}, \dot{\phi} \leq \dot{\phi}_{max}, \dot{v} \leq a_{max}, \dot{\psi} \leq \psi_{max}, v \leq v_{max}, v_h \leq v_{h,max}$$

(3. Wind)

Related motion in airmass to ground track – constrains the minimum radius of curvature.

$$\dot{x}_g = V_a \cos(\psi(t)) + V_{w,x}$$

$$\dot{y}_g = V_a \sin(\psi(t)) + V_{w,y}$$

Coordinated Turn Equations

Turning Radius

$$R = \frac{V_a^2}{g \tan \phi}$$

Center of Turn

$$c_n = p_n + R\cos(\psi - s\frac{\pi}{2})$$

$$c_e = p_e + R\sin(\psi - s\frac{\pi}{2})$$

Change in Angle

$$\Delta\phi = \frac{V_a}{R}\Delta t$$

New Position

$$p'_n = c_n + R\cos(\psi + s\frac{\pi}{2} - \Delta\phi)$$

$$p'_e = c_e + R\sin(\psi + s\frac{\pi}{2} - \Delta\phi)$$

V_a = Forward speed

R = Turning radius

ϕ = Roll angle

ψ = Heading angle

p_n, p_e = North, east position

c_n, c_e = Center of turn

Outline

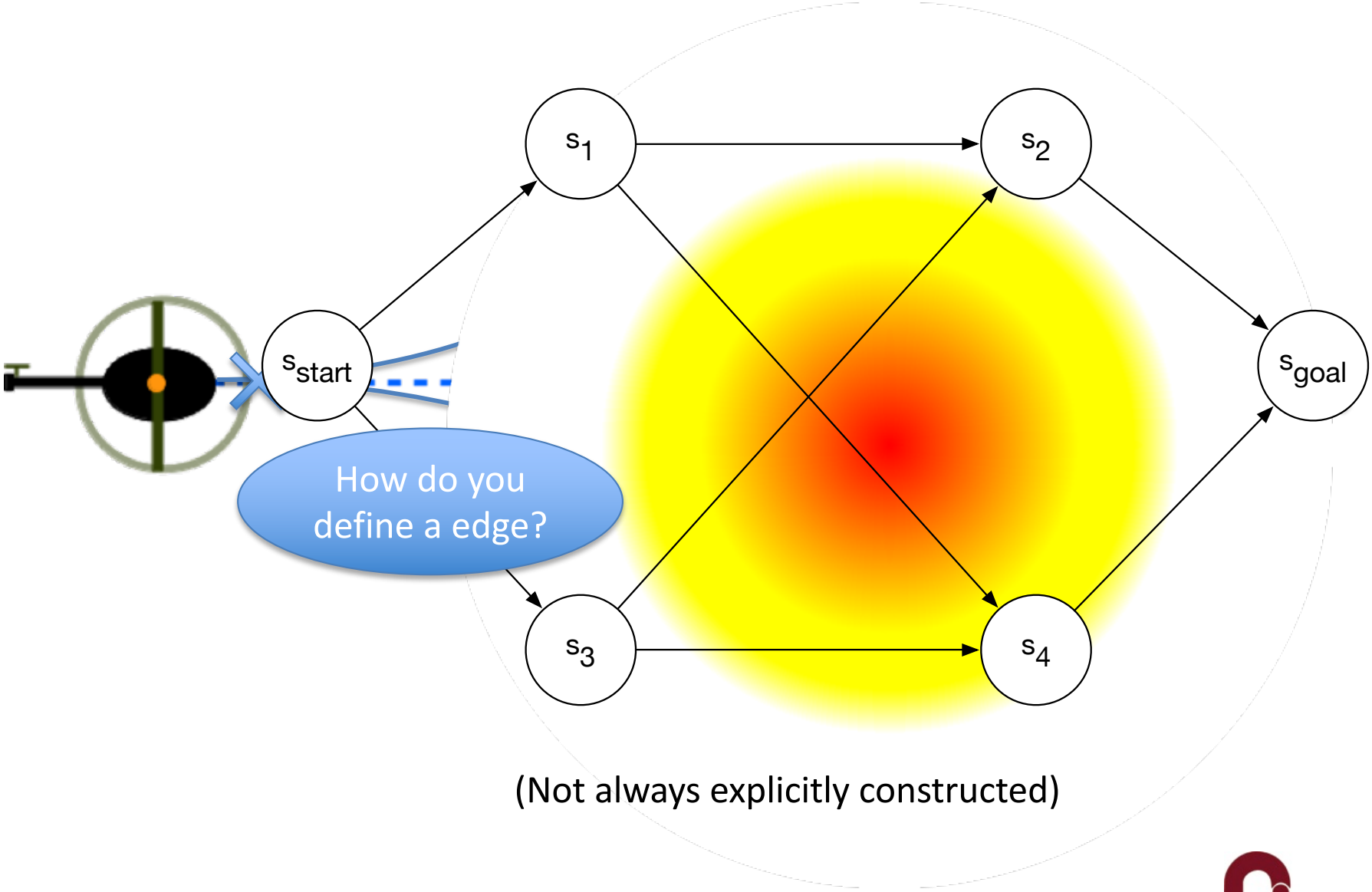
- Introduction
- Problem
- **Abstraction and Approach**
- Results
- Summary
- Exercise

The Trajectory Planning Problem

find	$\sigma(t) = \{x(t), y(t), z(t), \psi(t), t_f\}$	Time parameterized trajectory
minimize :	$J = \int_0^{t_f} c(\sigma(t))dt + c(\sigma(t_f))$	Cost function
constraints :	$\sigma(0) = \sigma_0$ $\sigma(t_f) = \sigma_f$	Boundary value constraints
	$h(\sigma(t), \dot{\sigma}(t), \ddot{\sigma}(t), \ldots) = 0$	Non-holonomic constraints
	$g(\sigma(t), \dot{\sigma}(t), \ddot{\sigma}(t), \ldots) \leq 0$	System limitations
	$J < \infty$	Cost function constraints

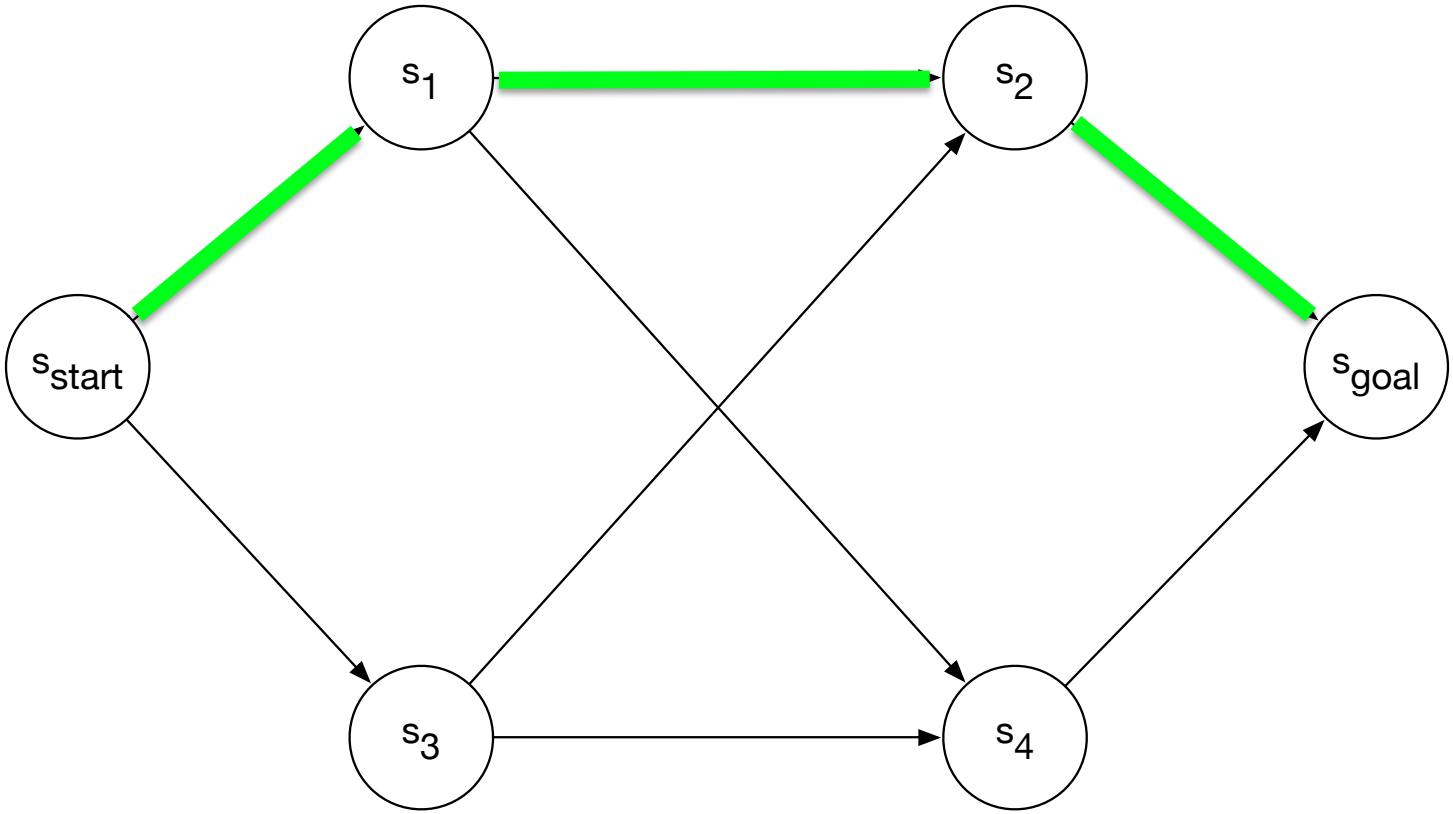
What are potential approaches to solve this problem?

How do we discretize the problem?



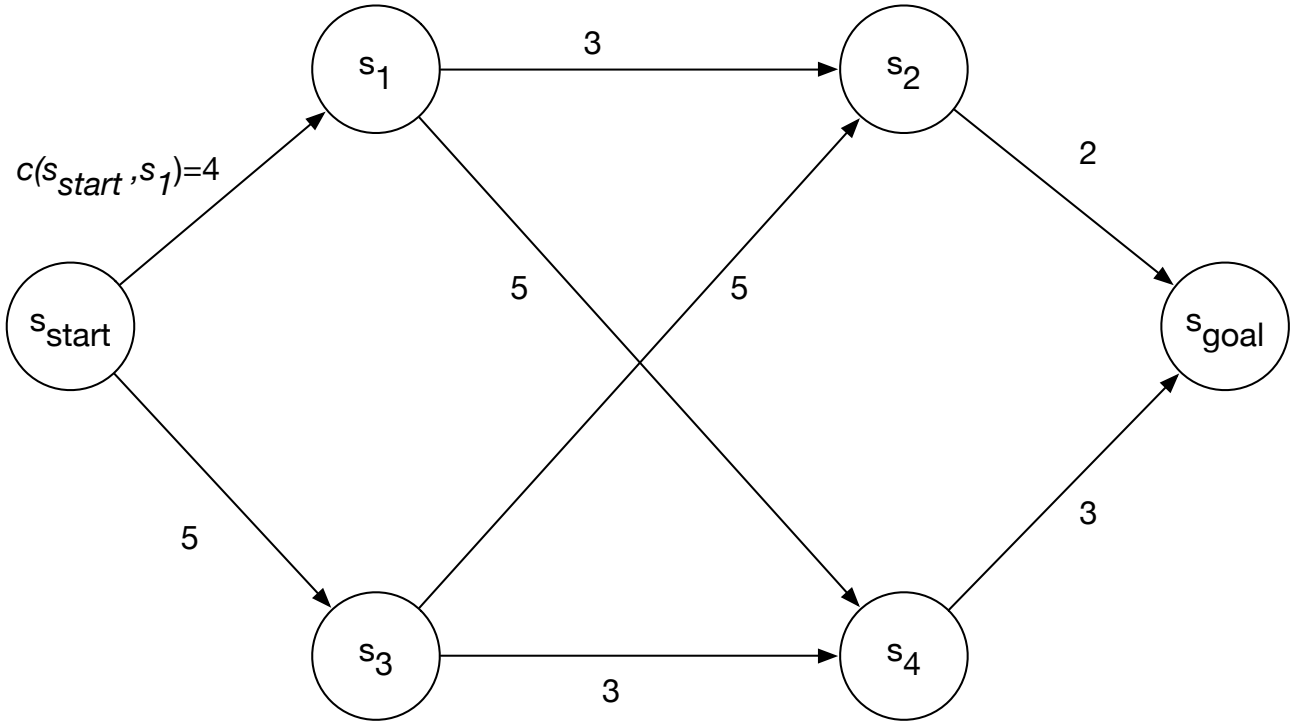
(Not always explicitly constructed)

How do we find the best path through the graph?



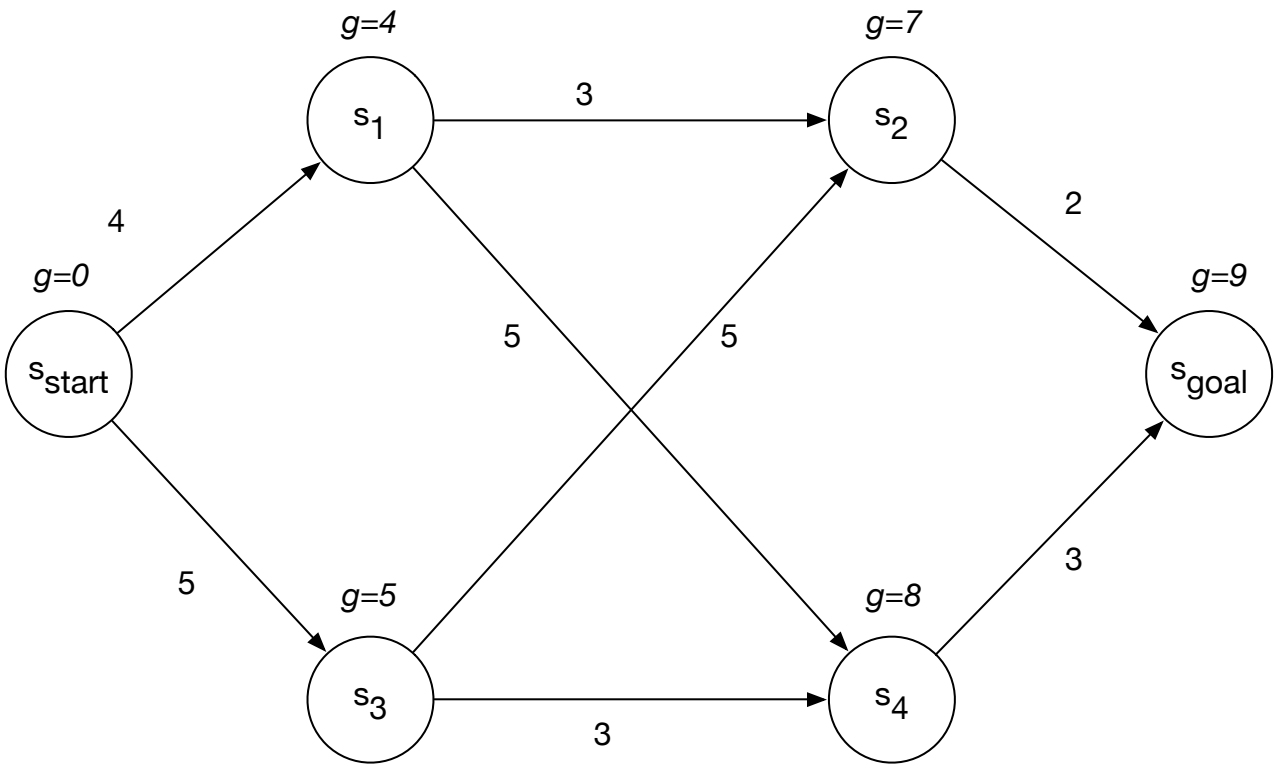
Edge Costs

- Cost between two nodes. (Can also have vertex costs but typically only edge.)
- Cost *c(s,s')* depends on the cost of the objective *J* of the trajectory segment of the edge *s->s'*
- Calculating *c(s,s')* can be expensive (collision checking)



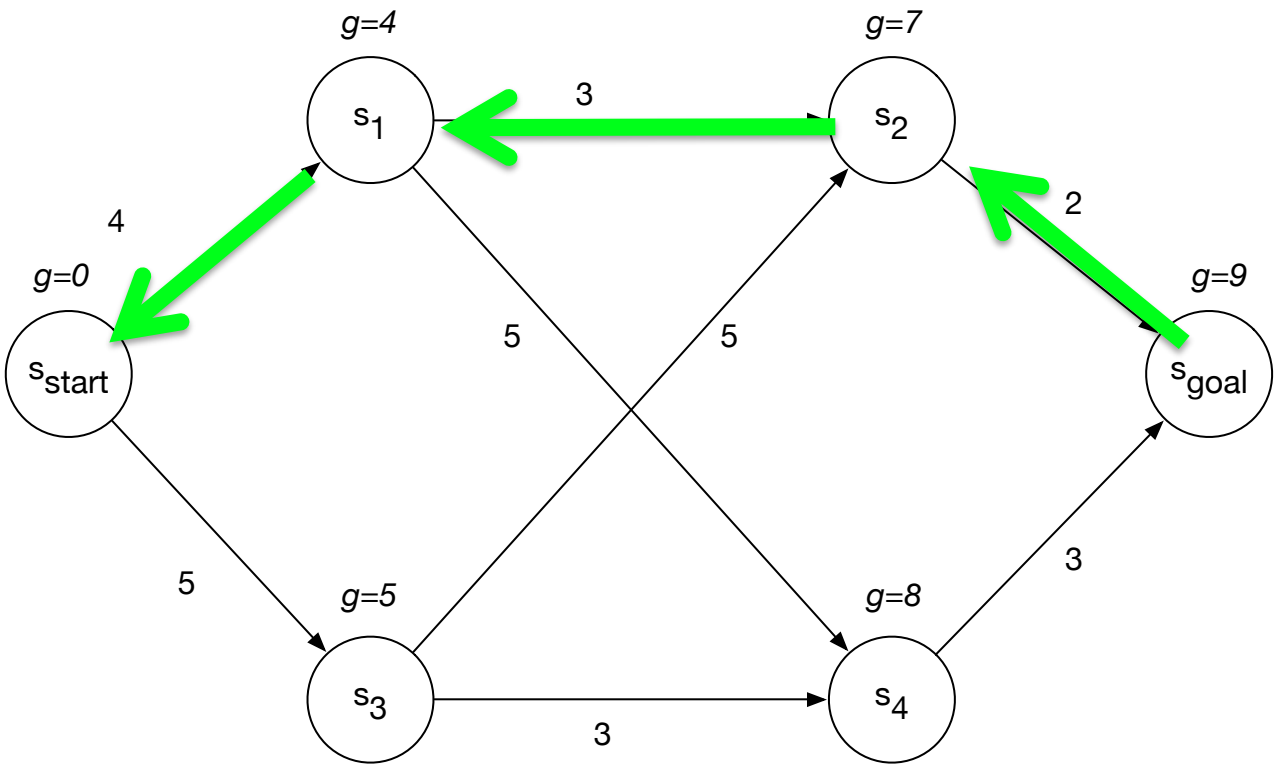
Calculating the least-cost path

g(s) cost of the least-cost path. Optimal *g* satisfies: $g(s) = \min_{s' \in pred(s)} g(s') + c(s', s)$



Finding the Least-Cost Path: Backtracking

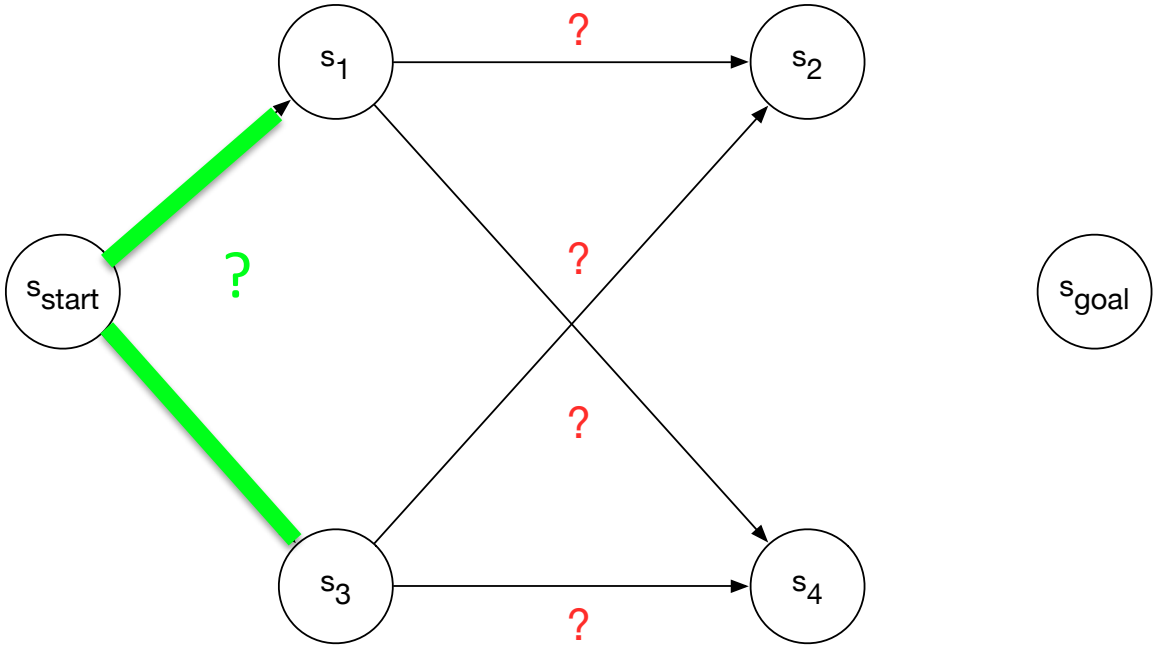
- Start at the goal and greedily backtrack to start: $s'' = argmin_{s' \in pred(s)}(g(s') + c(s', s)$



What are the main questions?

Fixed time/memory budget (real-time planning):

1. What vertices to create?
2. Where to search?



3 Representative approaches:

- Regular graph search: A*-grid search
- Sampling-based: RRT*
- Trajectory optimization: CHOMP

A*

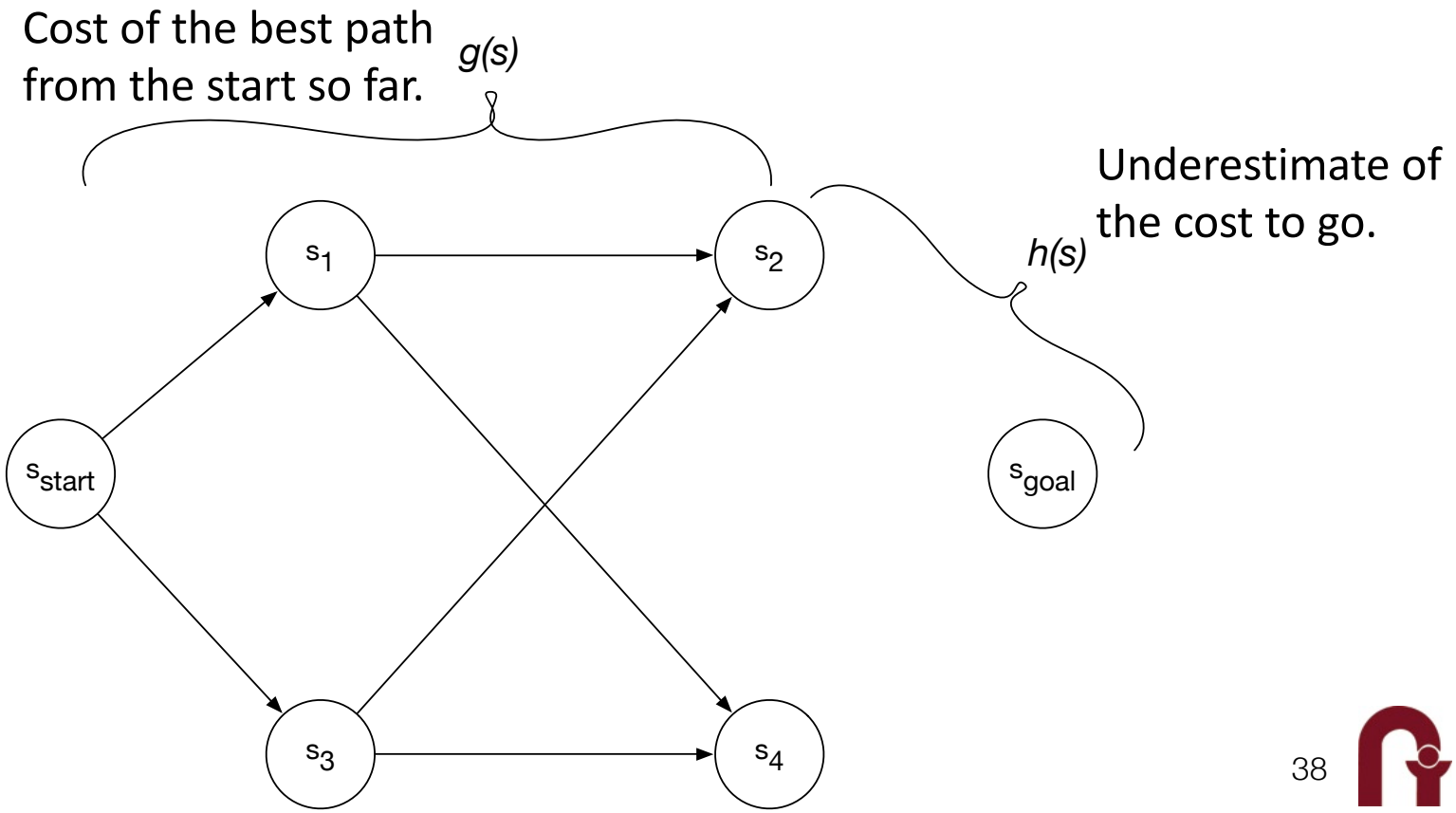
Compute optimal *g* and heuristic *h*

1. What vertices to create?

   Based on heuristic and cost *g+h*

2. Where to search?

   Look at priority queue



Admissible Heuristic Function *h*

- Popular function: Euclidean distance

h(s):

- Admissible: $h(s) \leqq c(s, s_{goal})$
- Consistent (satisfies triangle inequality):
  - $h(s_{goal}, s_{goal}) = 0$ and for all other states:
  - $h(s) \leqq c(s, succ(s)) + h(succ(s))$

A* Search

Update *g* based on the smallest g+h cost:

A:*

$g(s_{start})=0$; $g(s \neq s_{start})=\infty$; OPEN=$\{s_{start}\}$

while($s_{goal} \neq s$)

remove s with the smallest $g(s)+h(s)$ from *OPEN*

expand s

A* Properties

- Resolution-complete and optimal
- Minimum number of state expansions

Questions

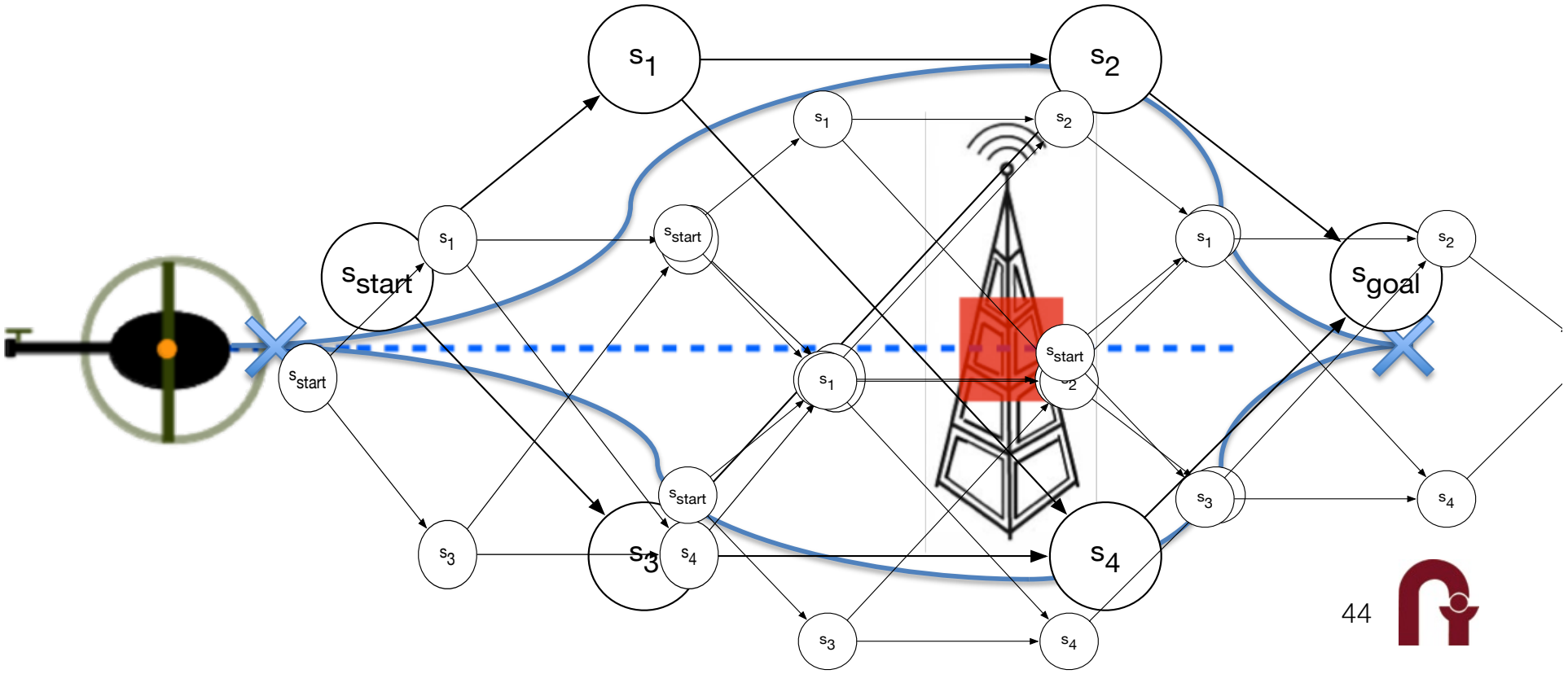
- What graph should you create?
- What resolution to pick?
- How do you expand it?
- Performance depends on
  - Graph: Branching factor/Abstraction of the system
  - Quality of the heuristic for the system and environment
  - Match of the graph abstraction to the environment

Going Deeper

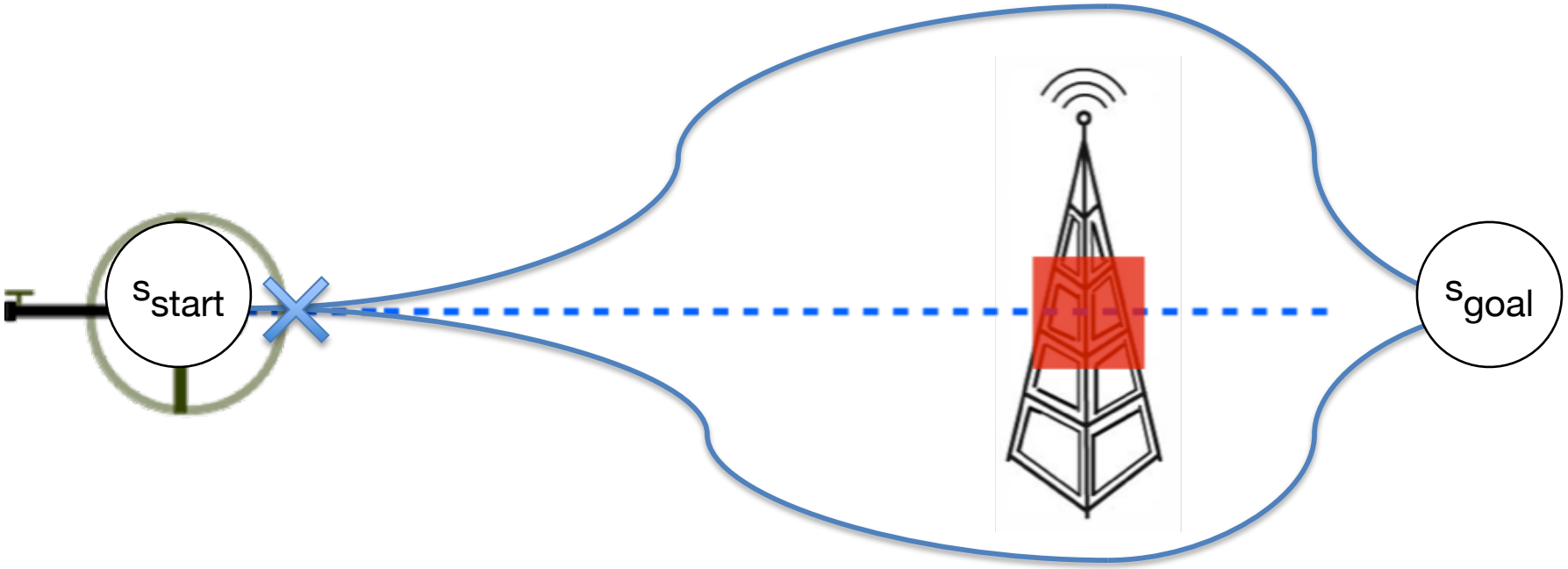
- How do you incorporate dynamics into your graph?
  - State lattice: [Pivtoraika09]
  - Maneuver Automaton: [Frazzoli02]
- What are more interesting heuristics for dynamical systems?
  - Precompute heuristics [Knepper06]
  - Dubin's heuristic [Dubins57]
- How do we repair rather than redo the search for the changing graph?
  - D* Lite [Koenig02]
  - Anytime D* [Likhachev05]
  - Anytime Search [Hansen07]

Sampling Based Planning

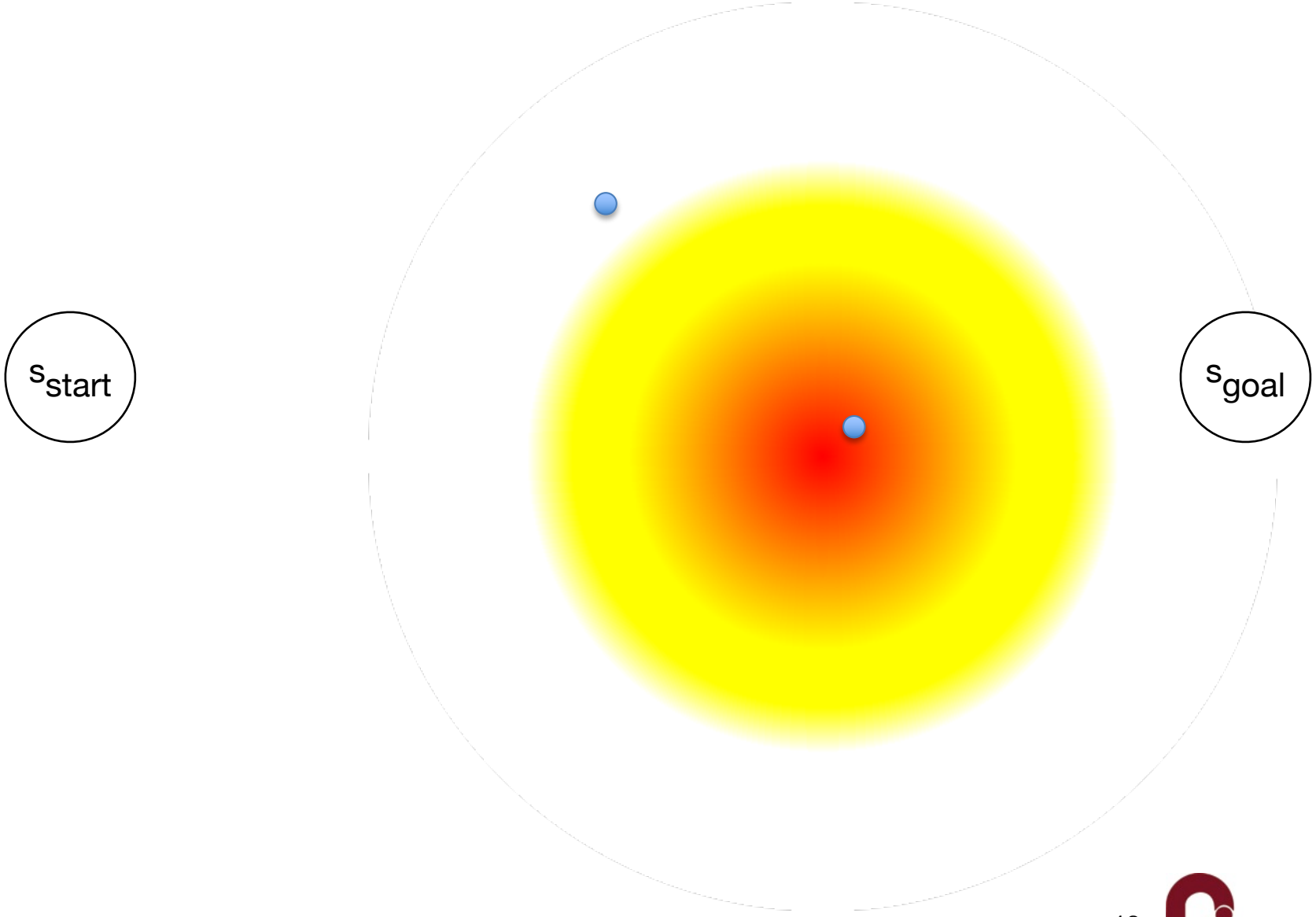
- Where should you put your graph?
- What resolution to pick?



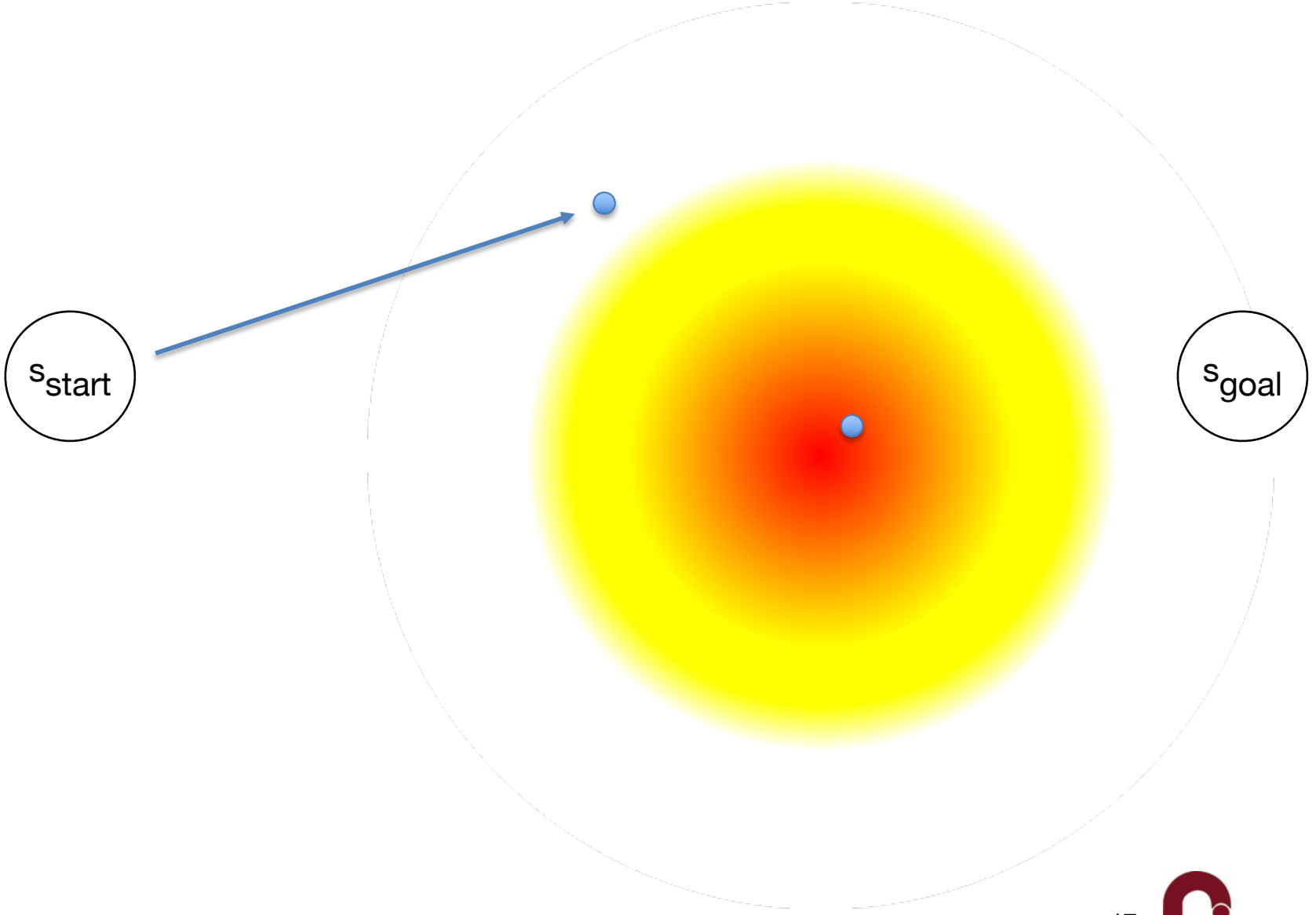
Sample the Environment (in an Increasingly Denser Fashion) and Connect the Samples to Construct a Tree to Find a Path to the Goal (RRT*)



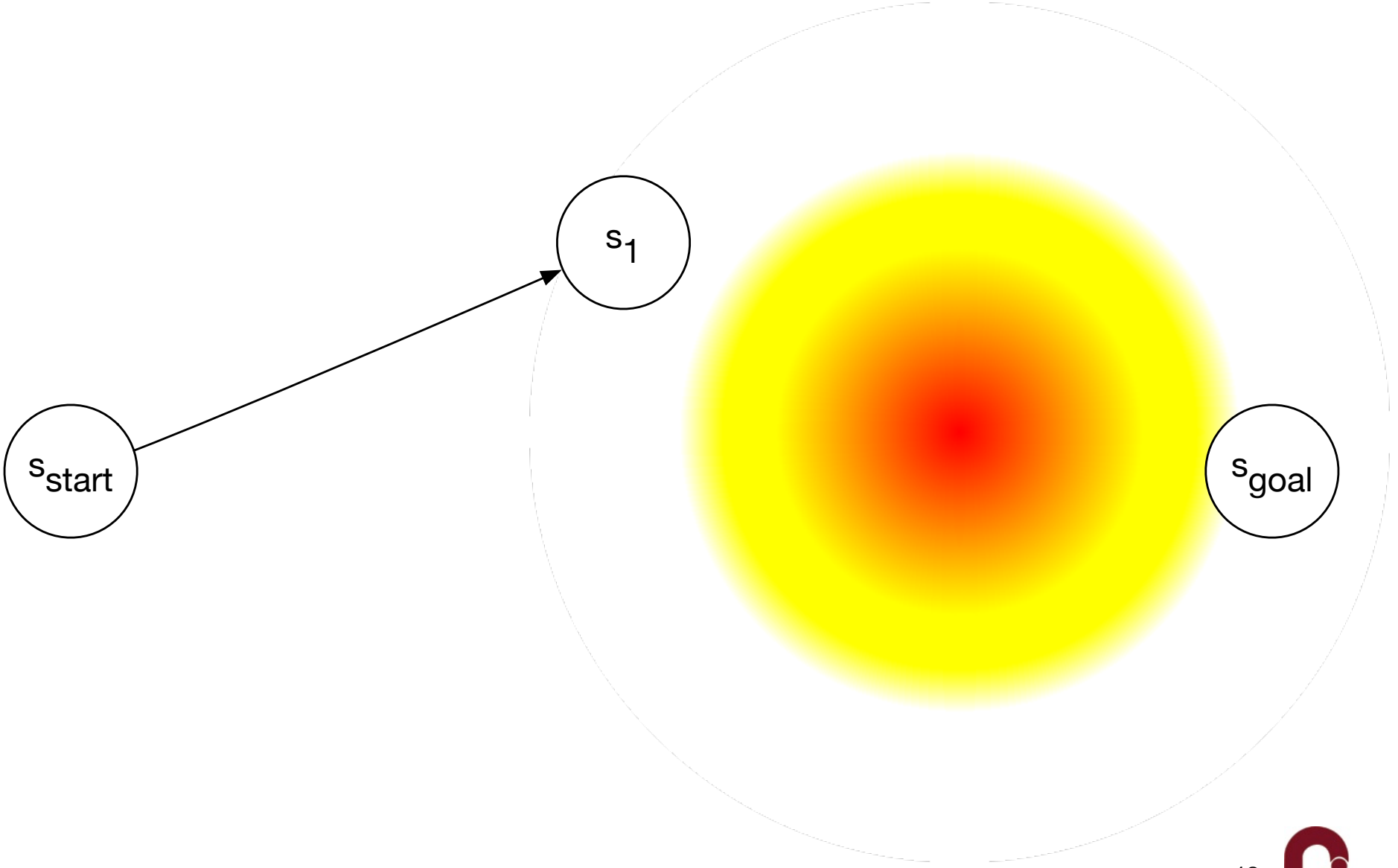
Sample a Potential Location to Expand To



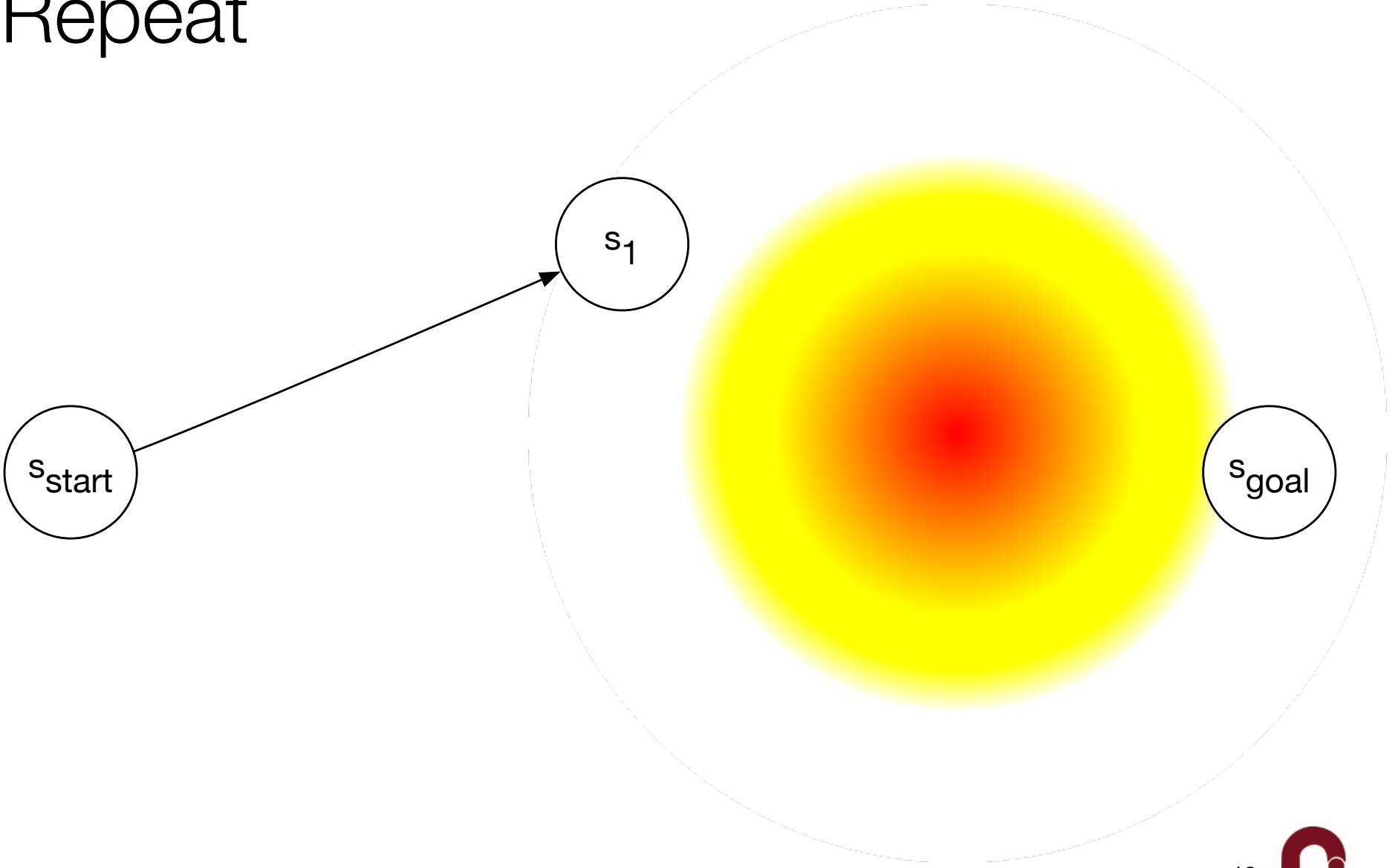
Add Edge to Connect to Graph



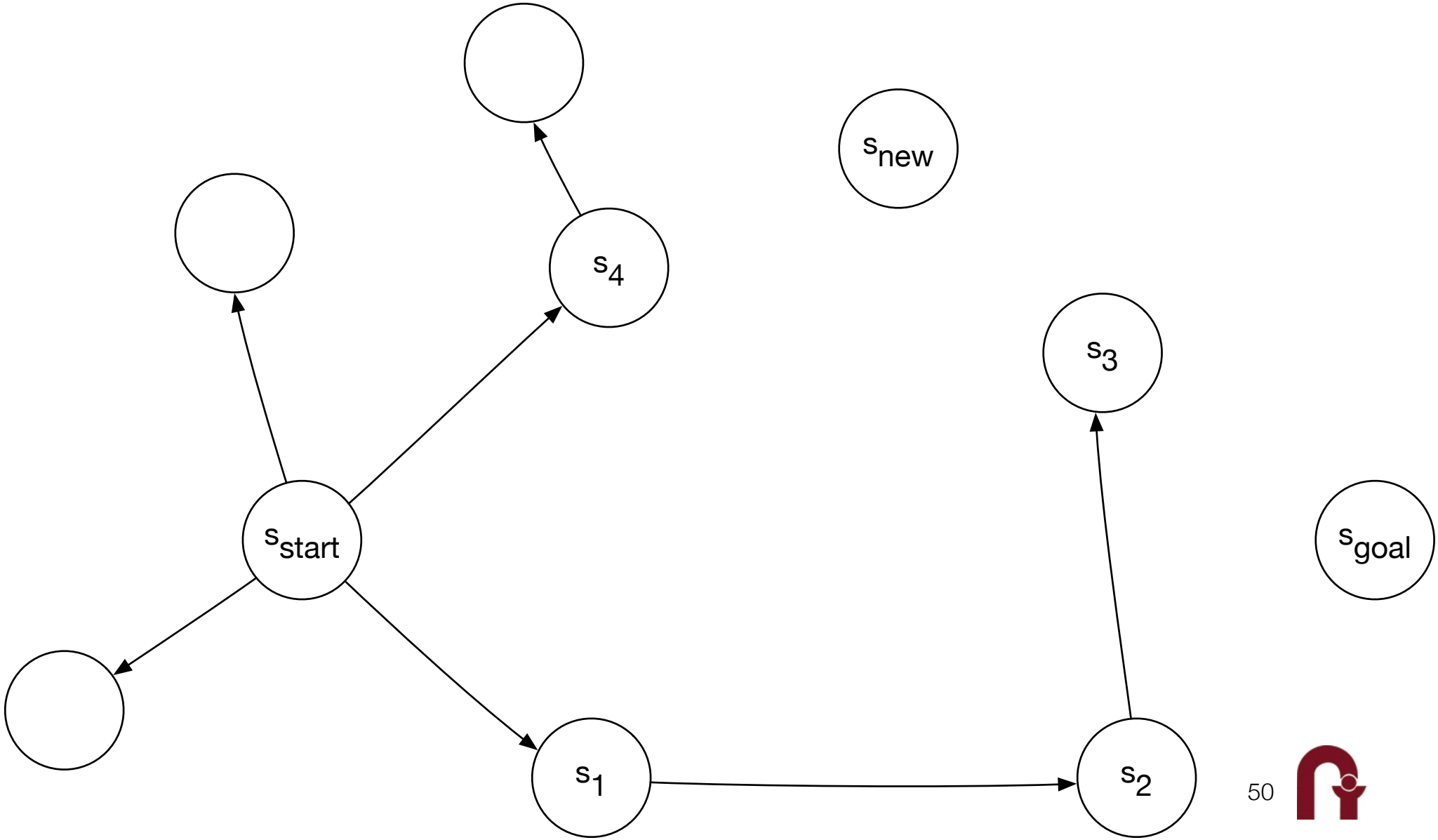
Add Edge to Connect to Graph



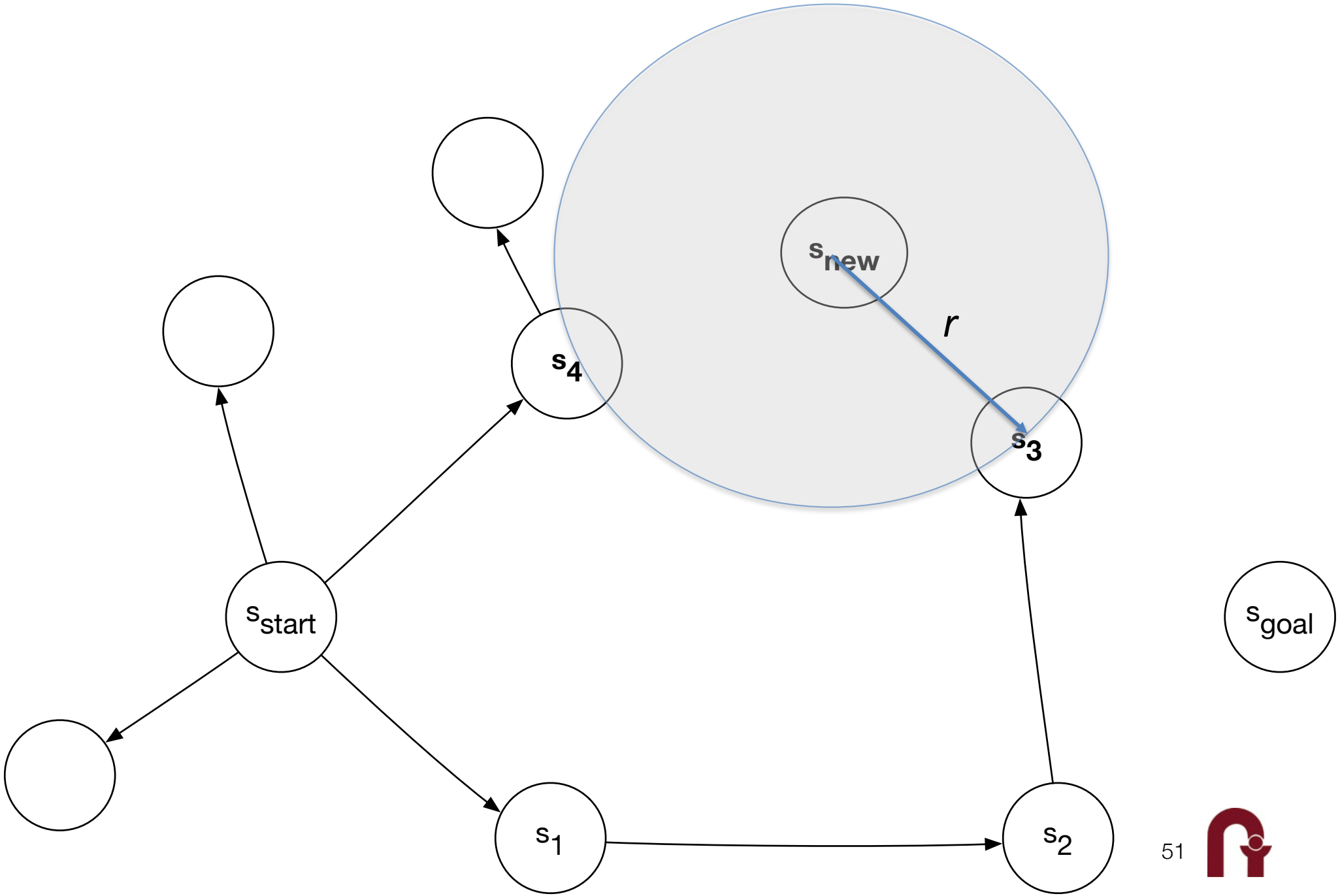
Add Edge to Connect to Graph and Repeat



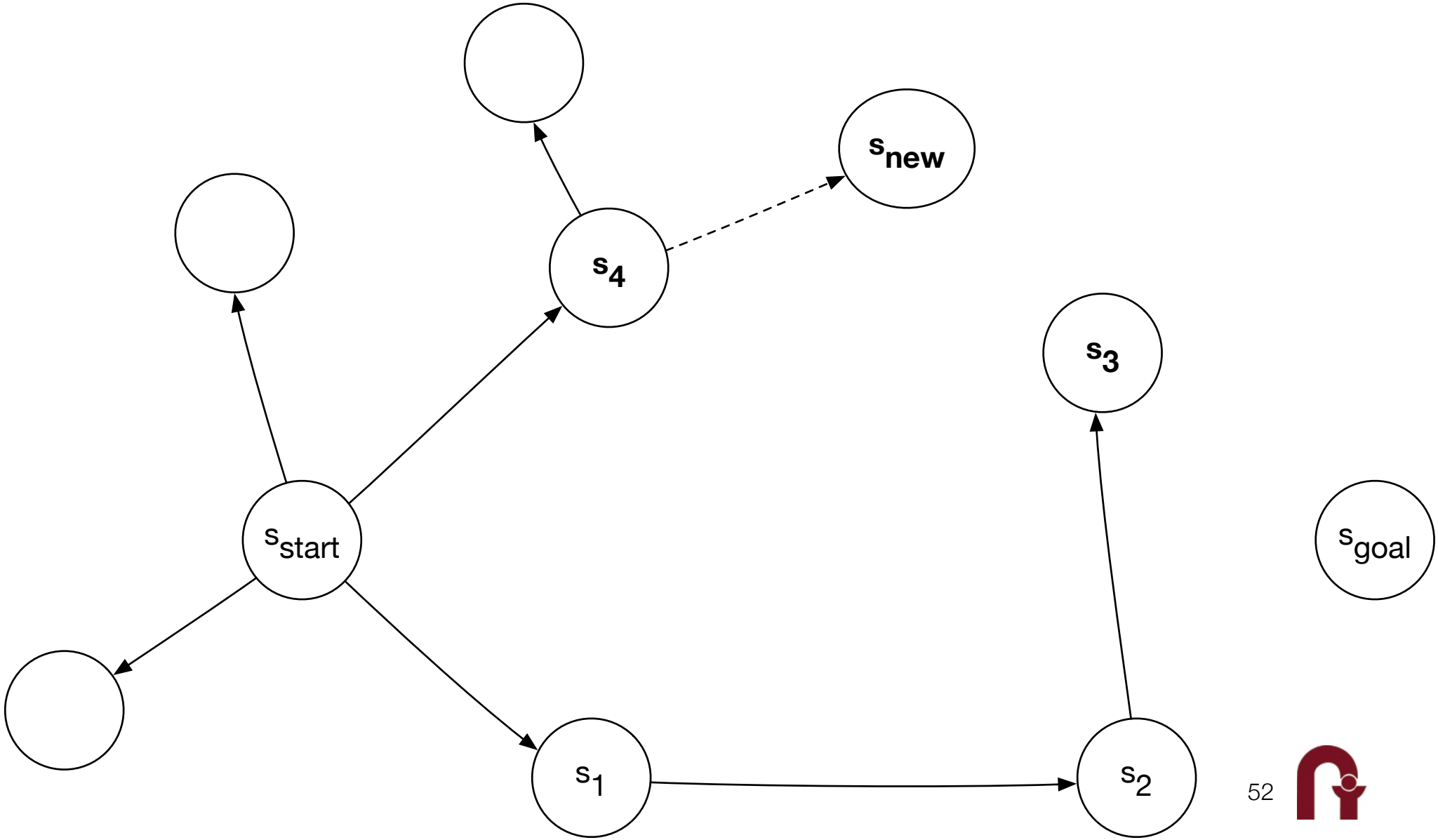
Some Iterations Later…



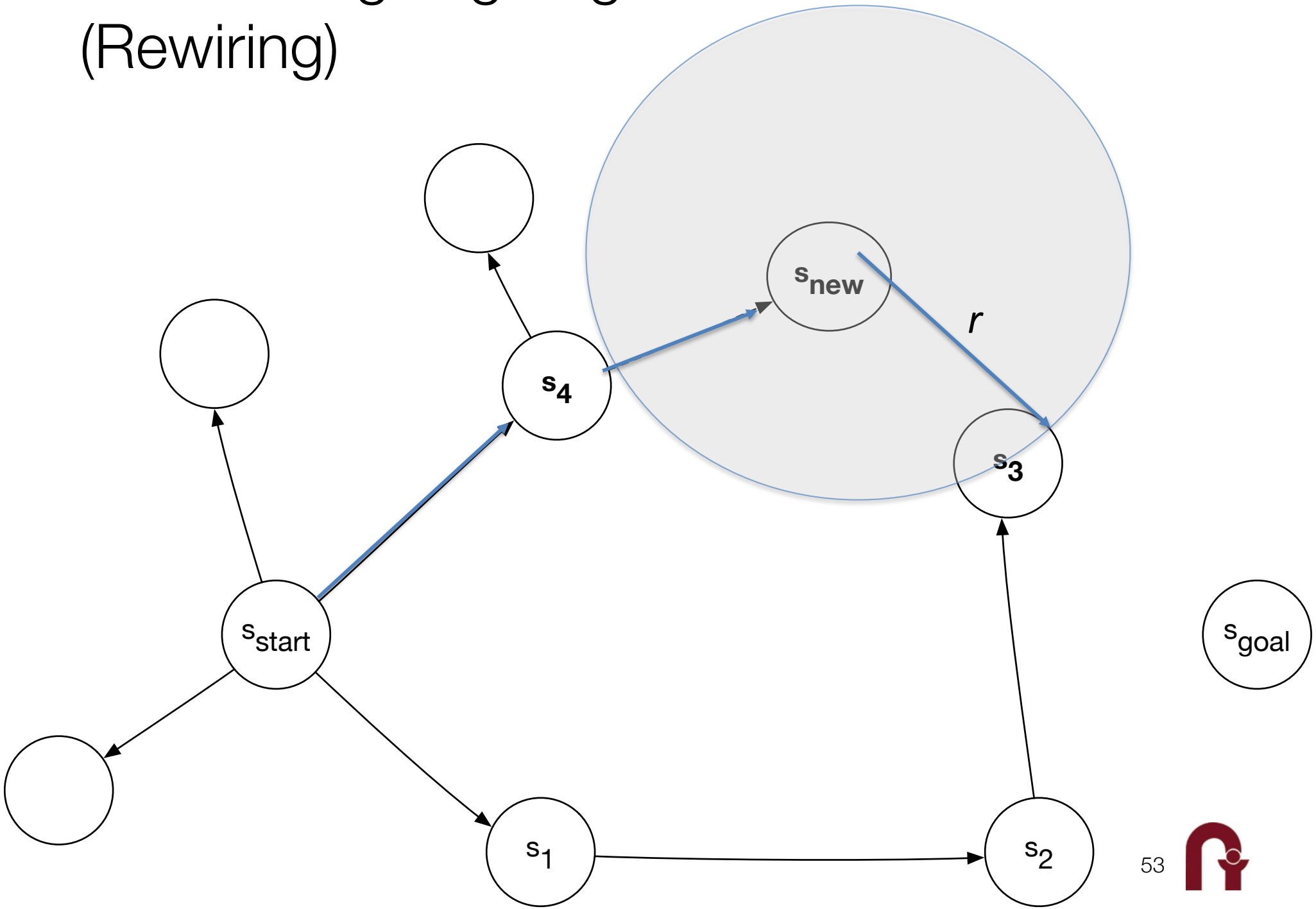
Look at the neighbors within *r*



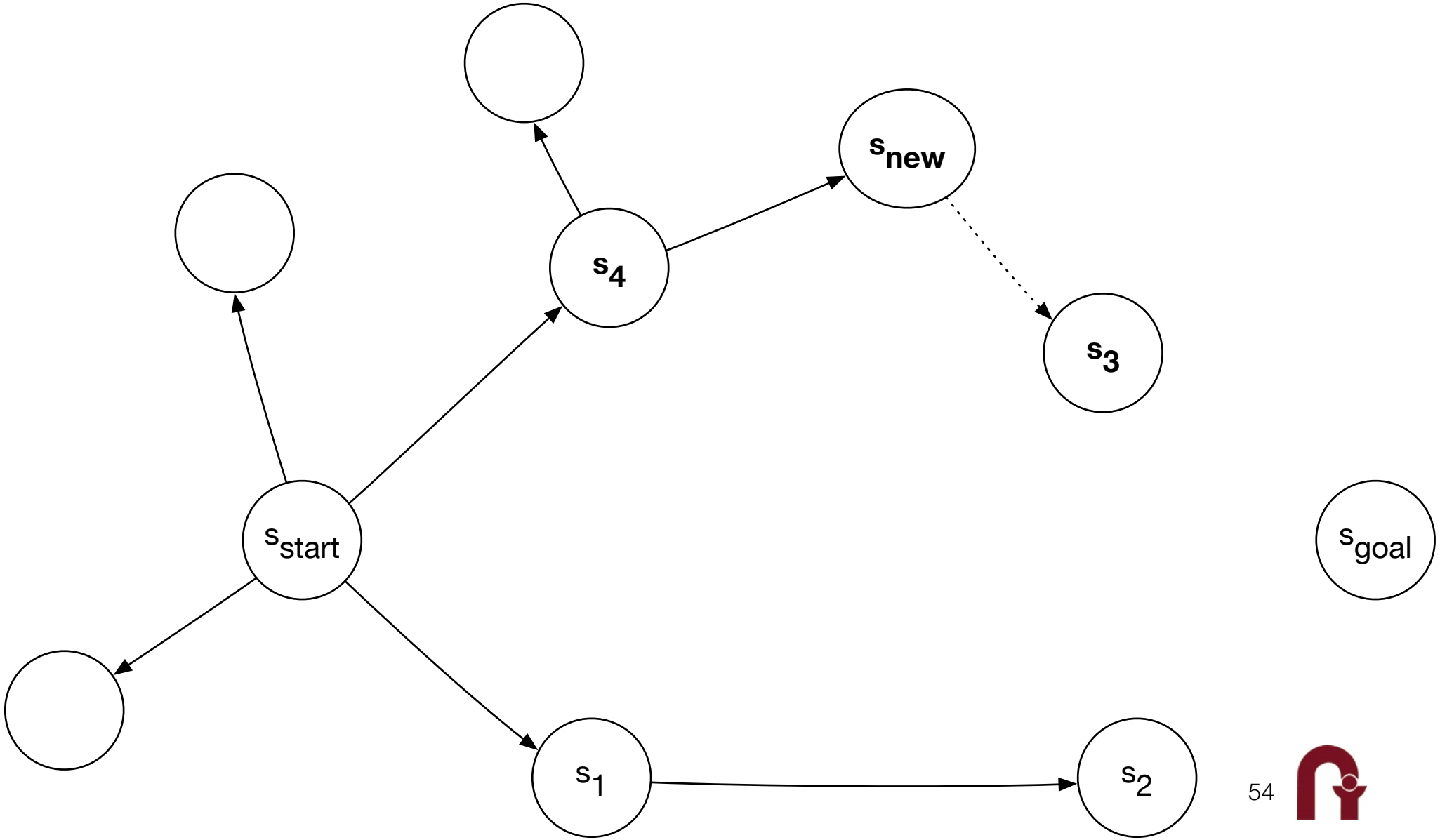
Connect



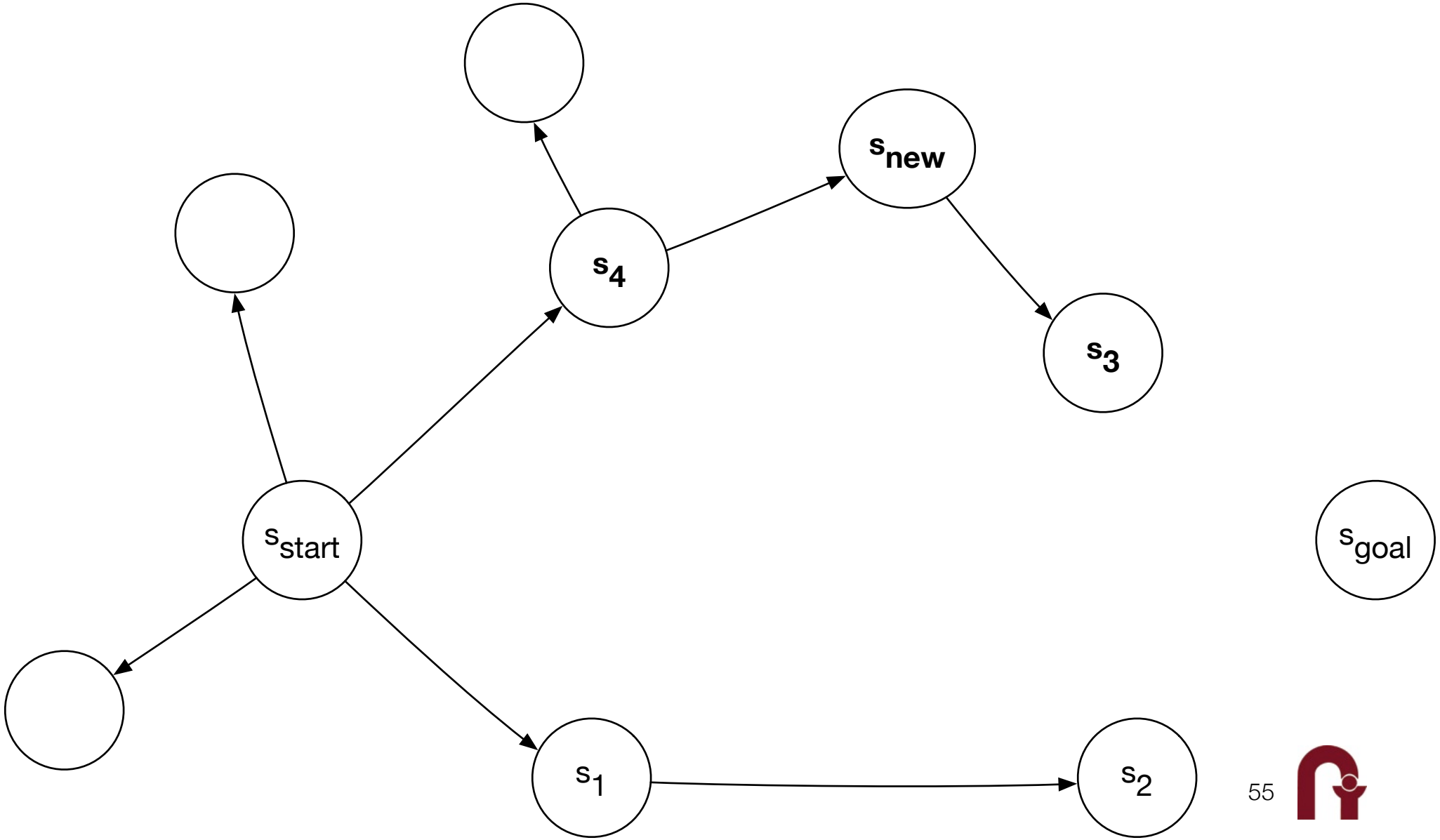
Check outgoing edges for lower costs (Rewiring)



Change the Parent and Fix Tree



Change the Parent and Fix Tree



RRT* Algorithm (Concept)

$V = s_{start}$; $E = \{\}$

for i=1…n

s_{new} = getNewValidRandomSample()

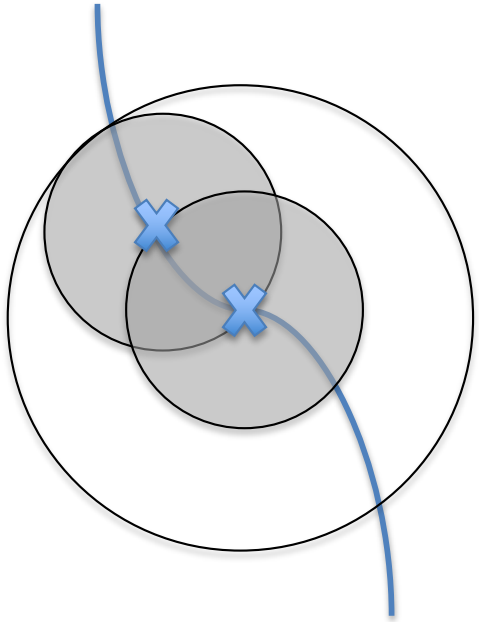
S_{near} = getVerticesWithin($r(i)$)

(s_{min}, J_{min}) = getLowestCostNeighbor(S_{near})

$E = E \cup (s_{min}, s_{new})$, $S = S \cup s_{new}$

E = rewireTree(E, S_{near})

Radius



card(V)= number of vertices, *d* = number of dimensions

$\mu(X_{free})/\zeta_d$ Ratio of the volume to the volume of the unit sphere.

$$r = \gamma_{RRT^*} \left(\frac{\log\left(card(V)\right)}{card(V)} \right)^{1/d}$$

$$\gamma_{RRT^*} = 2\left(1 + \tfrac{1}{d}\right)^{1/d} \left(\mu(X_{free})/\zeta_d\right)^{1/d}$$

Why rewire the tree?

- Remove unnecessary detours
- Optimize for the minimum cost rather than committing to connections to early.

Questions

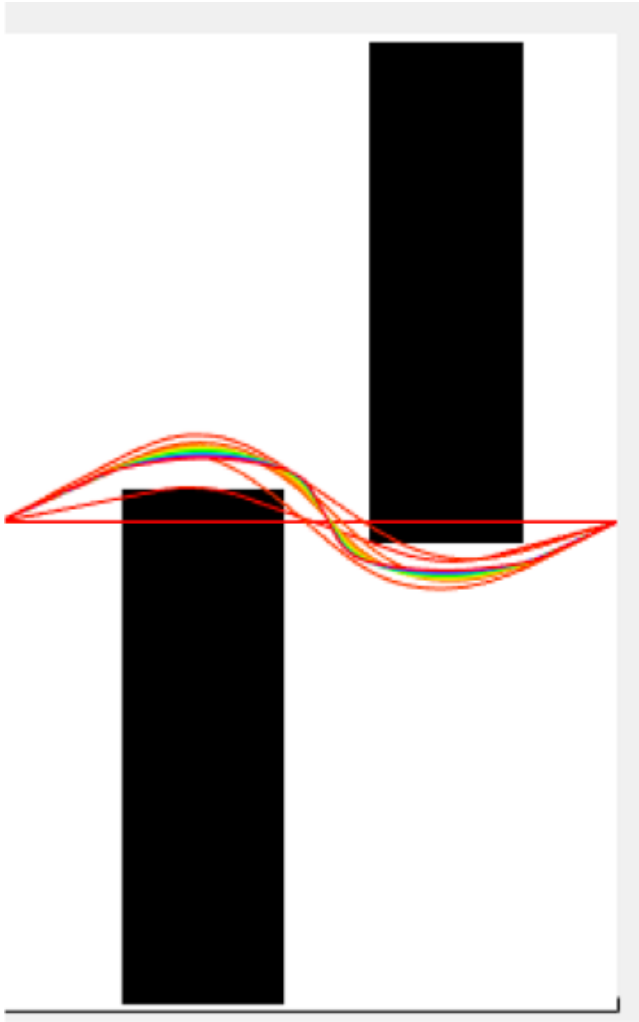
- What is expensive in this algorithm?
- What about r if our robot is motion constrained?
- How can one incorporate heuristics?
- In what environments will this algorithm perform well?

Going Deeper

- Why does this particular choice of *r* lead to optimum plans? [Karaman11]
- What if we consider a batch of samples to expand and keep a heuristic? [Gammell15]
- What if we one to have alternative routes instead of just the best route? [Choudhury13]

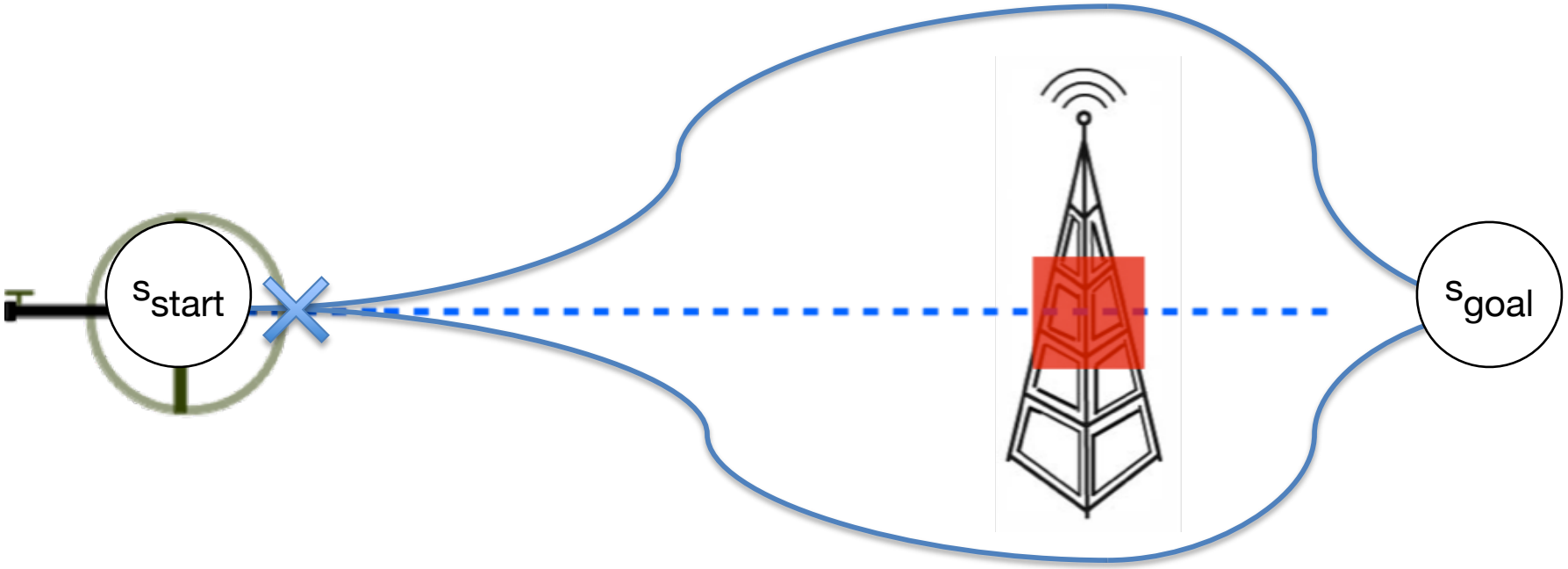
Trajectory Optimization (CHOMP)

- Exploit the first order information available about the trajectory
- Perturb an initial guess to minimize the cost function
- Example on left:
  - Straight line initial guess
  - Several optimization steps

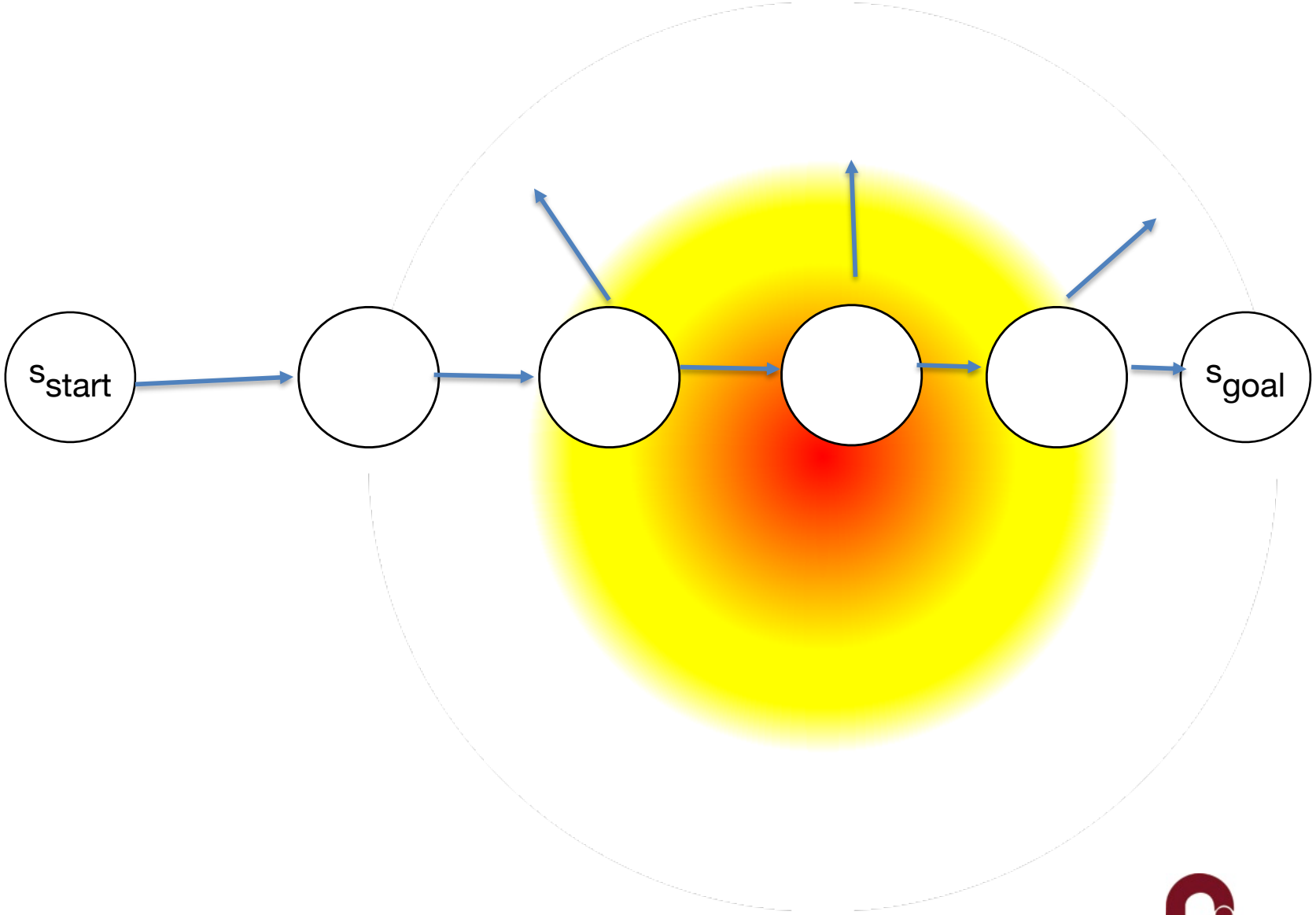


Zucker13

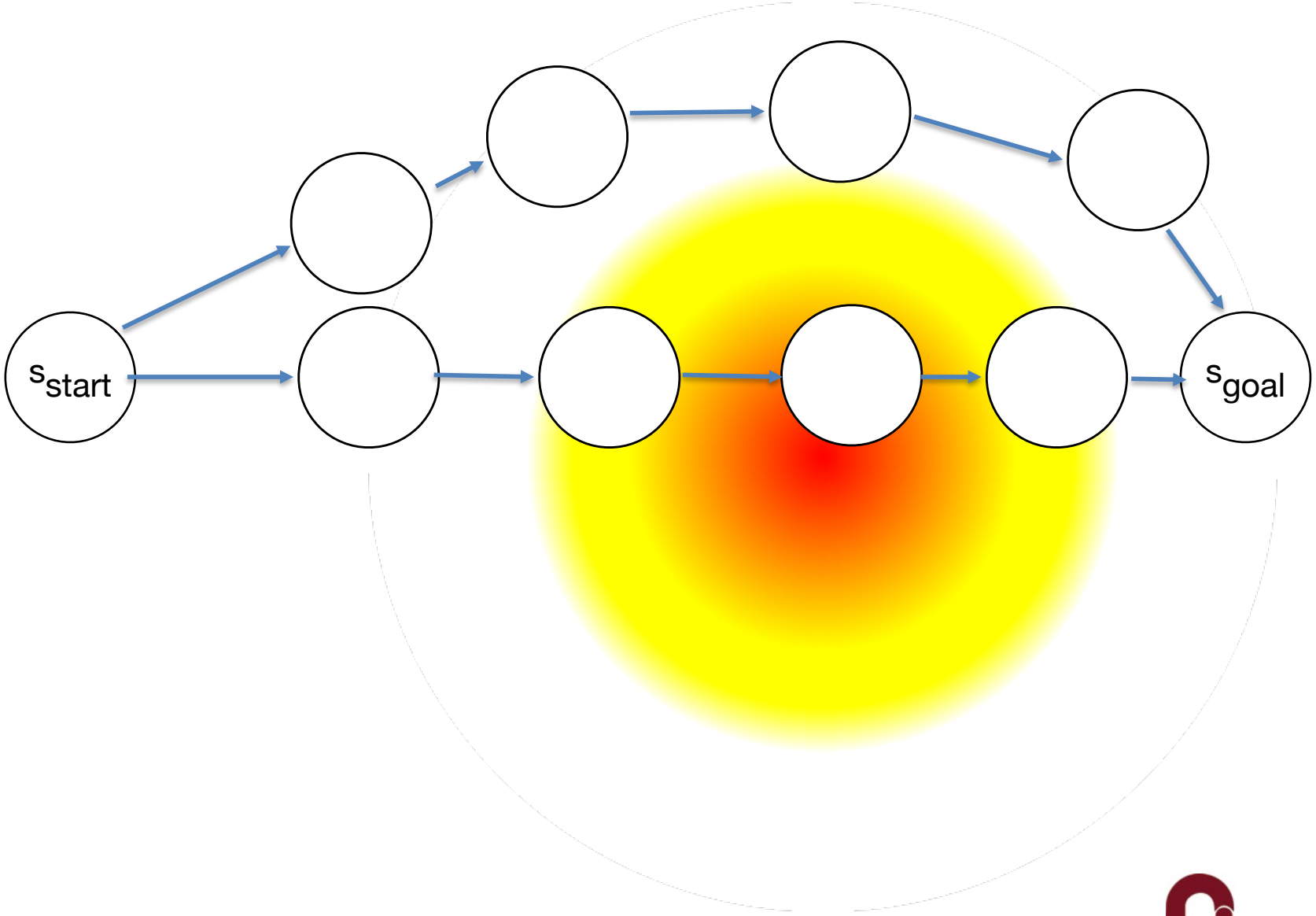
Going Back To Our Graph Example:



Add Vertices based on the Gradient



Add Vertices based on the Gradient



Cost Function

- Want the trajectory to be smooth to be executable by the robot and reach the goal

$$\mathcal{F}_{smooth}[\xi] = \frac{1}{2}\|K\xi + e\|^2 = \frac{1}{2}\xi^{\mathrm{T}} A\xi + \xi^{\mathrm{T}} b + c$$

- Avoid obstacles

$$\mathcal{U}[\xi] = \mathcal{F}_{obs}[\xi] + \lambda\mathcal{F}_{smooth}[\xi]$$

Key Idea

- Minimize the update to the trajectory with a smooth perturbation of the trajectory
- For example minimize the amount of velocity or acceleration added

=> Perform steepest descent in trajectory space

$$\xi_{k+1} = \xi_k - \frac{1}{\lambda} M^{-1} g_k$$

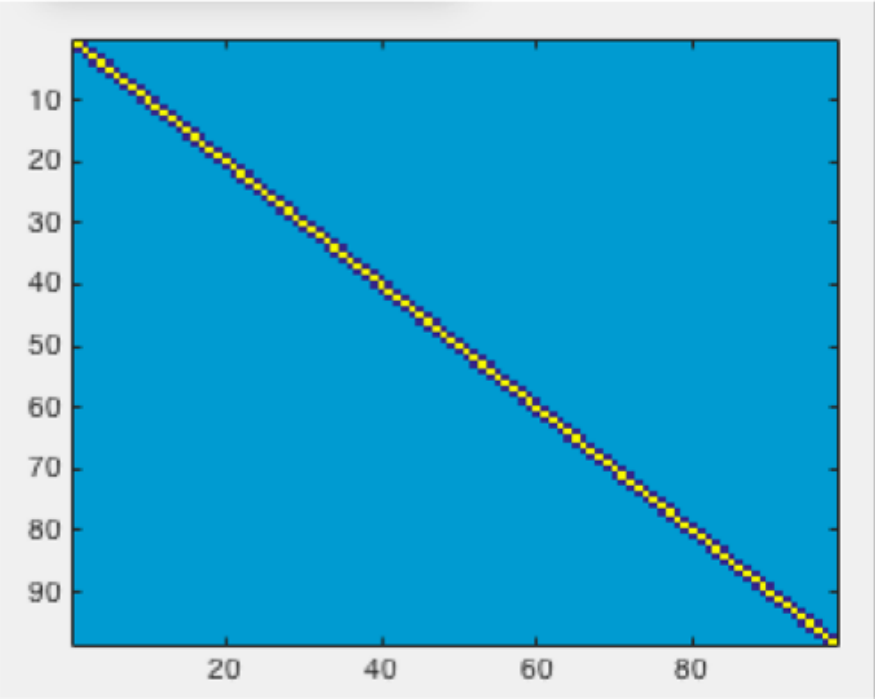
Measuring the Difference in Trajectory Space

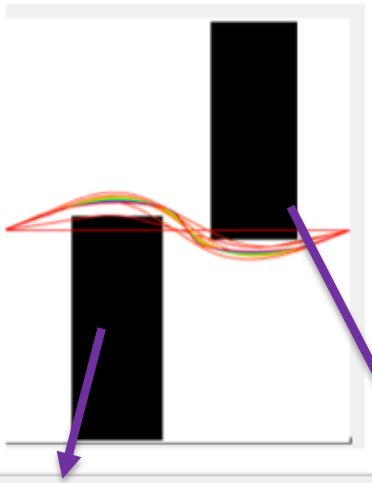
Depends on your representation.

Example M for waypoints:

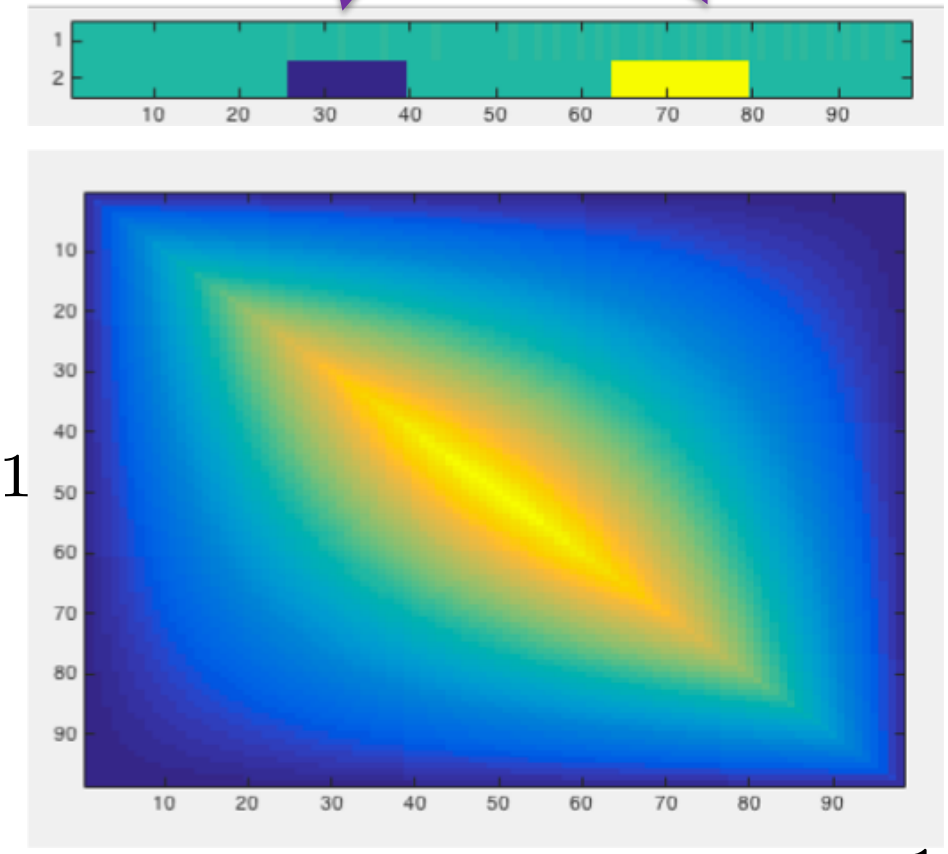
Finite difference between waypoints

M



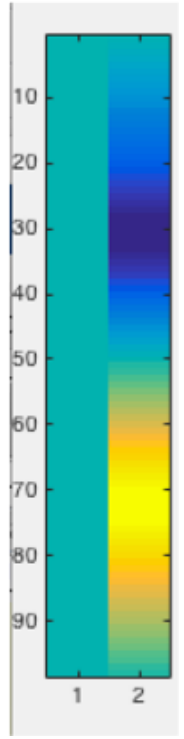


g_k



M^{-1}

$$\frac{1}{\lambda} M^{-1} g_k$$



$$\xi_{k+1} = \xi_k - \frac{1}{\lambda} M^{-1} g_k$$

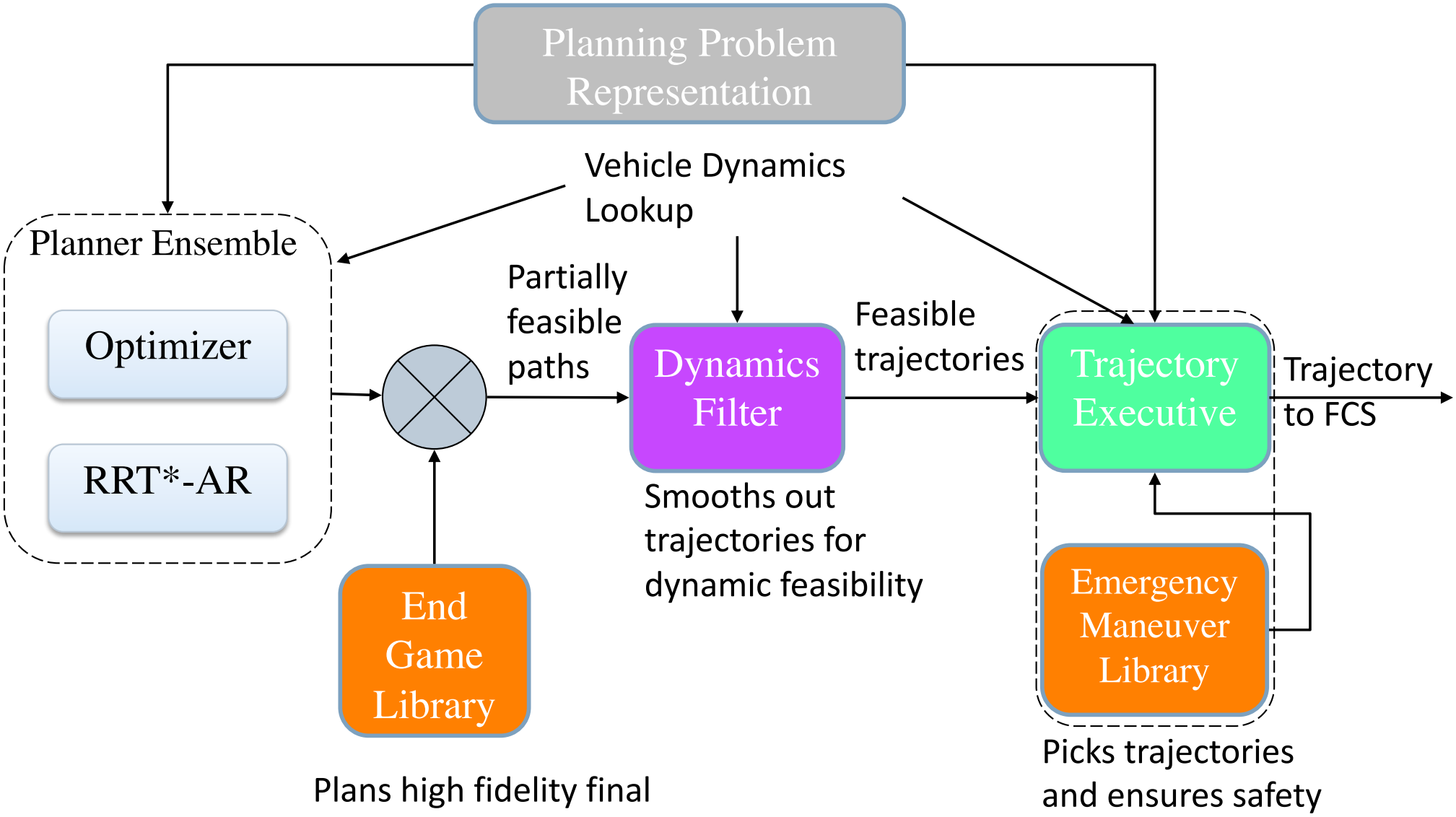
Questions?

- Is it complete and optimal?
- How could one include dynamic constraints in the trajectories (other than projection)?
- Why is it an effective method for air vehicles?
- How could you include other planning approaches in the optimization?

Going Deeper

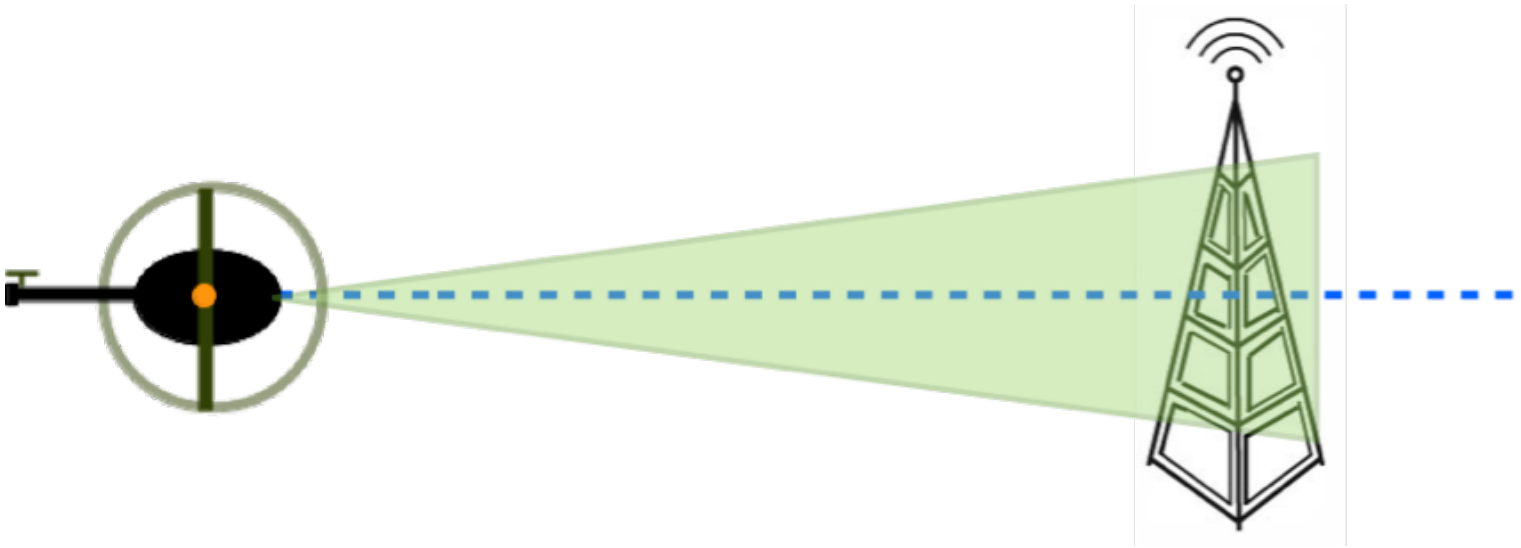
- Enforcing constraints on the trajectory? Project back along same space
- More challenging or larger environments? [He2013]
- Other trajectory optimization approaches? [Kolter09]
- Combing RRT* with Optimization? [Choudhury16]

Planner Ensemble Idea



Choudhury et al. AHS Forum 2014

Conceptual Illustration of Trajectory Planner

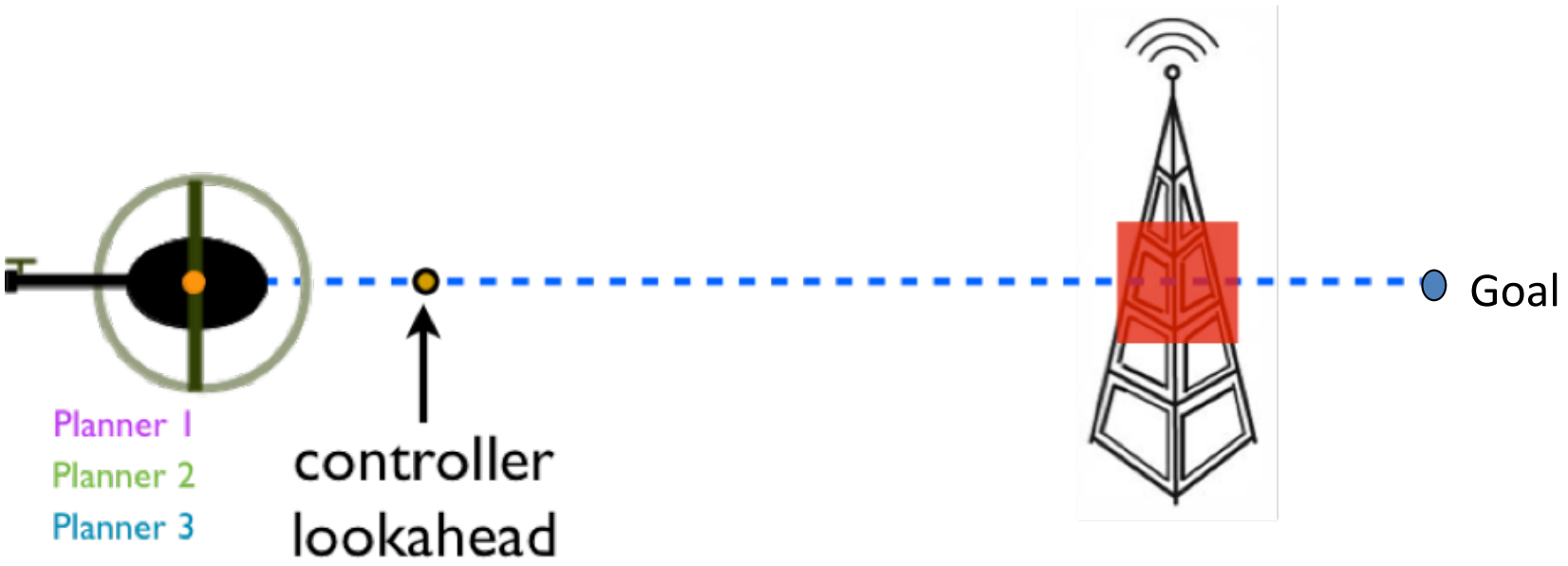


Obstacle enters sensor range

Conceptual Illustration of Trajectory Planner

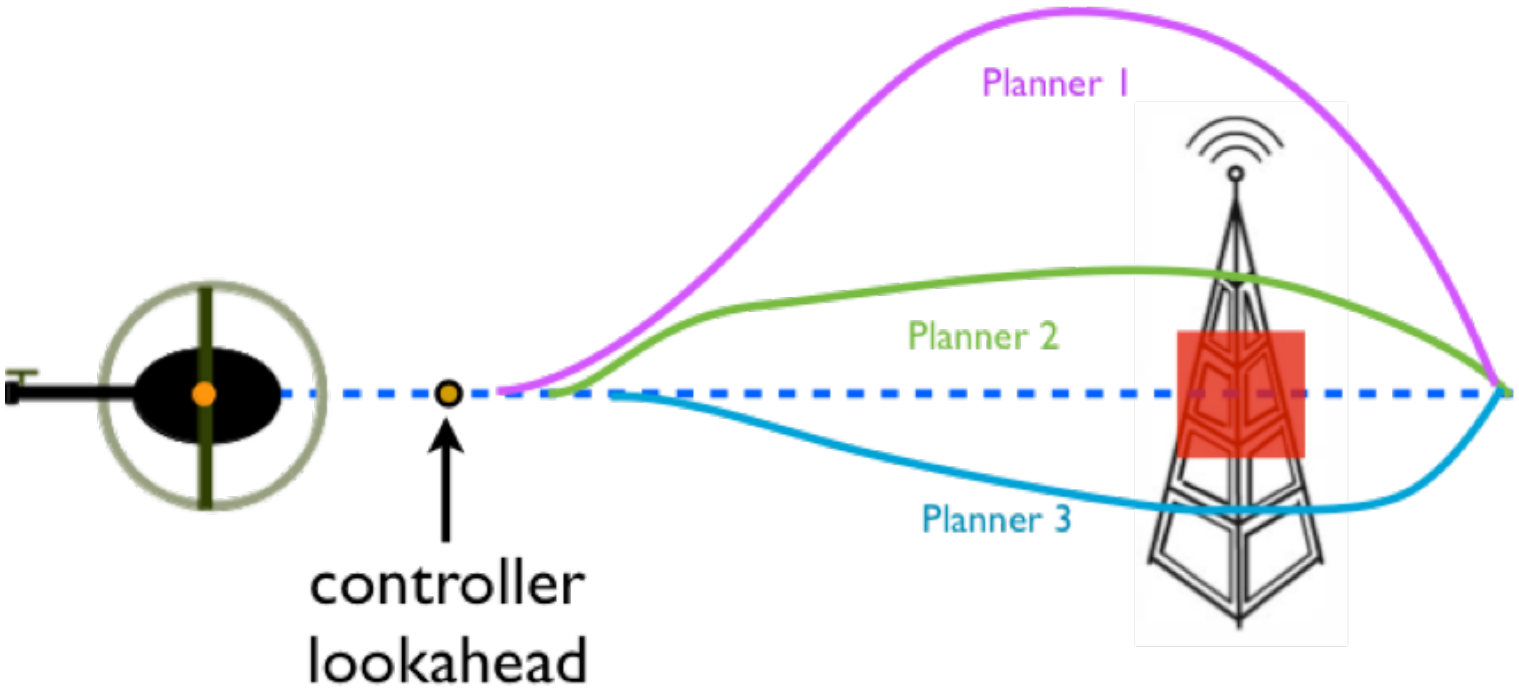
Obstacle mapping updates occupancy grid

Conceptual Illustration of Trajectory Planner



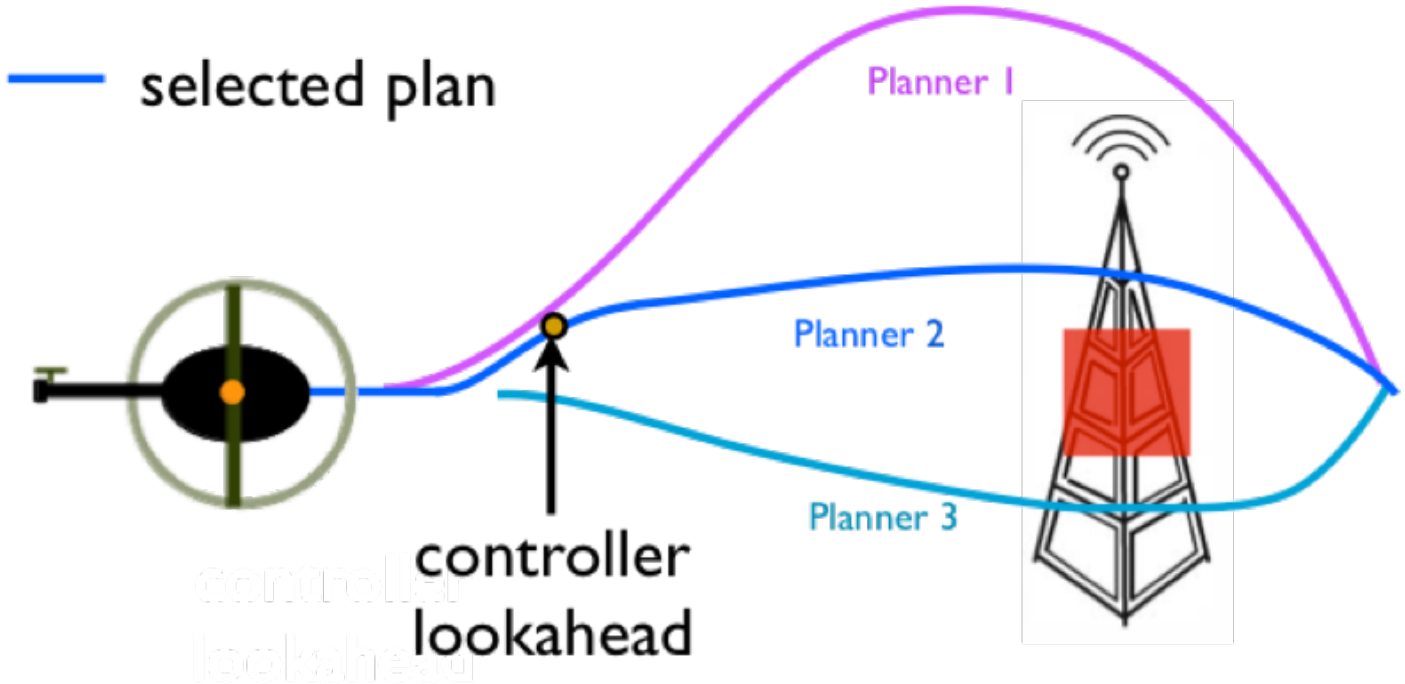
Planners receive start pose (lookahead) and goal pose

Conceptual Illustration of Trajectory Planner



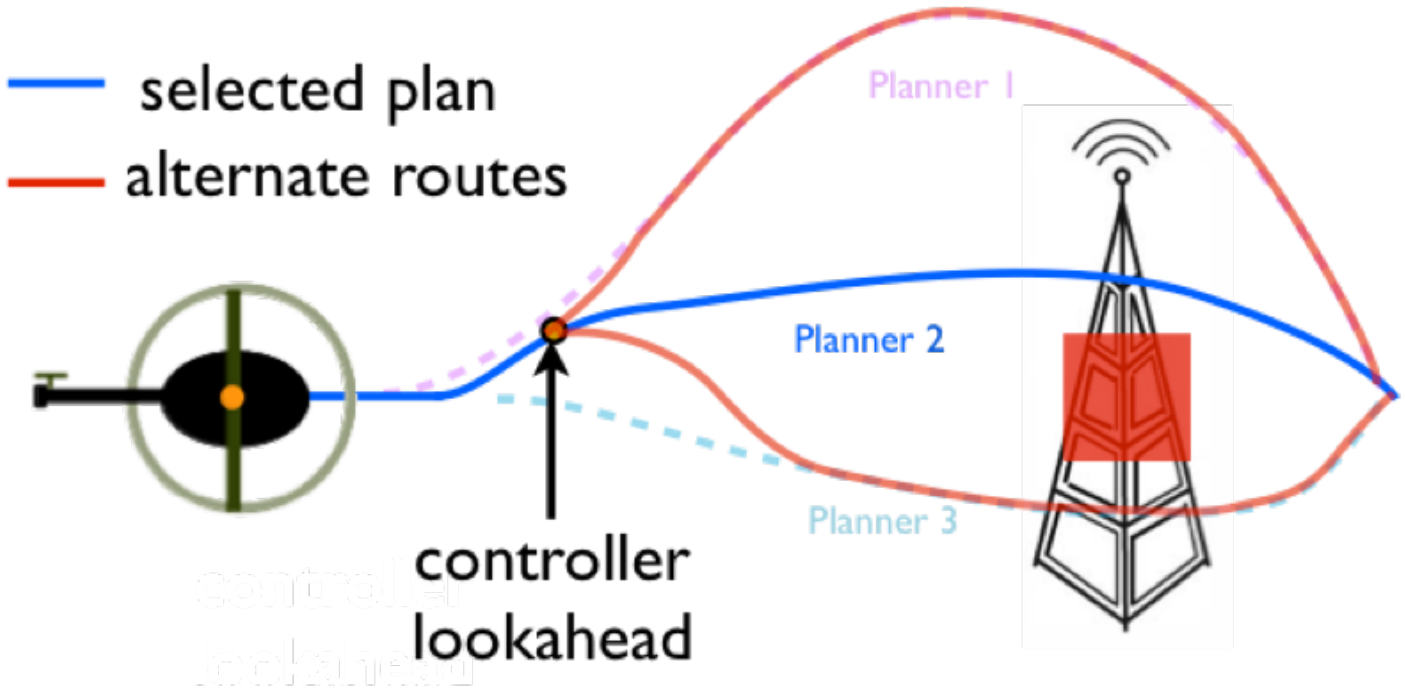
Each planners computes a trajectory and gives it to the executive

Conceptual Illustration of Trajectory Planner



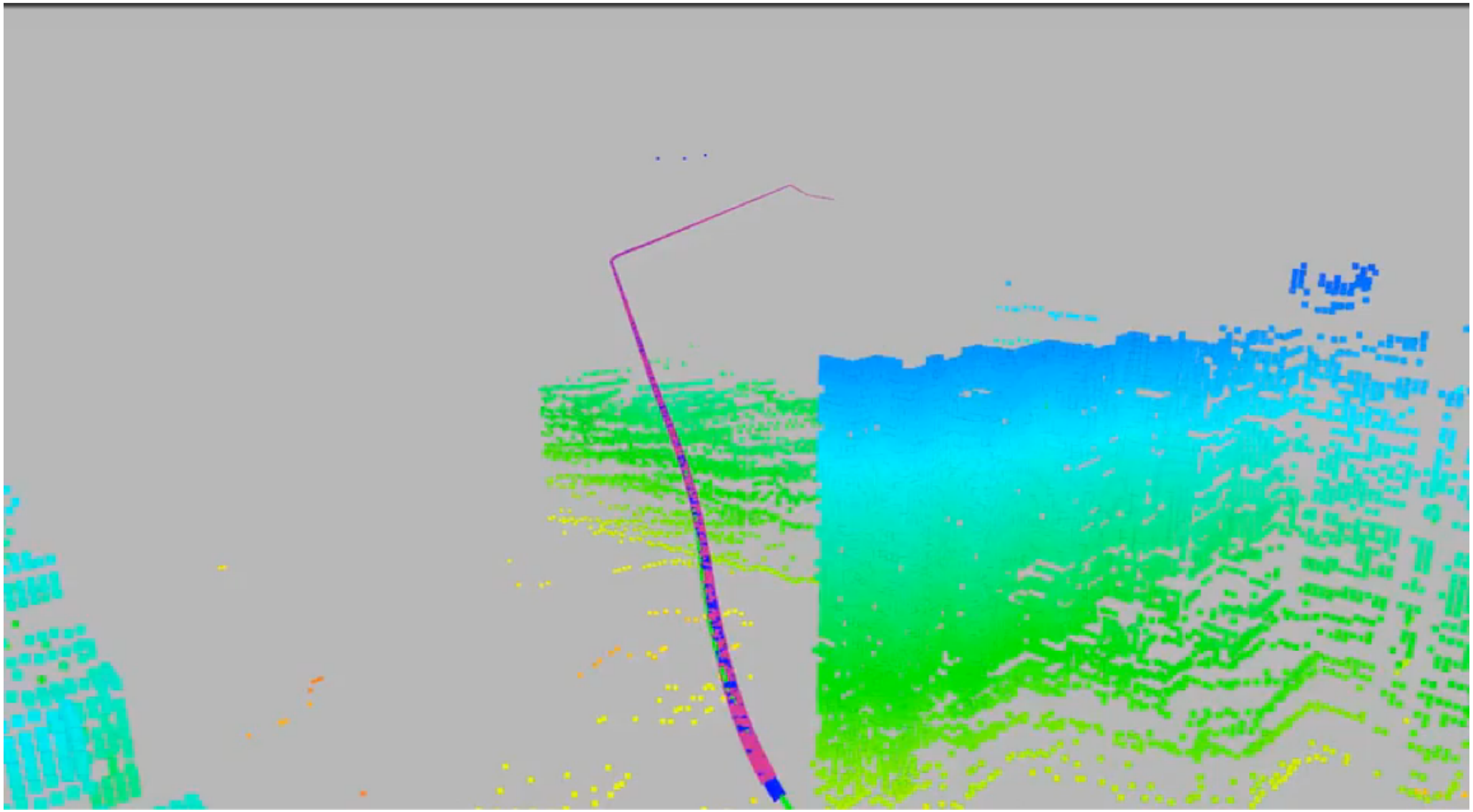
Executive selects planner 2 trajectory as optimal path

Conceptual Illustration of Trajectory Planner



Executive retains other paths as alternate routes

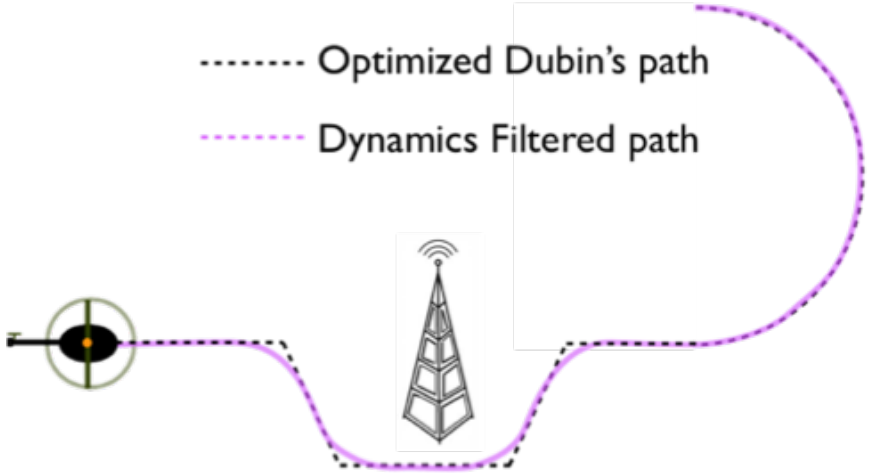
Ensemble/Executive result



Dynamics Filter – Producing feasible paths

Dynamic's Filter

- Accepts a partially feasible trajectory and filters it to be fully dynamically feasible.
- Dynamics have non-linear constraints – no analytical solution exists
- Thus planners plan dubin's curve (analytic) which is filtered to be within a funnel around the original path.



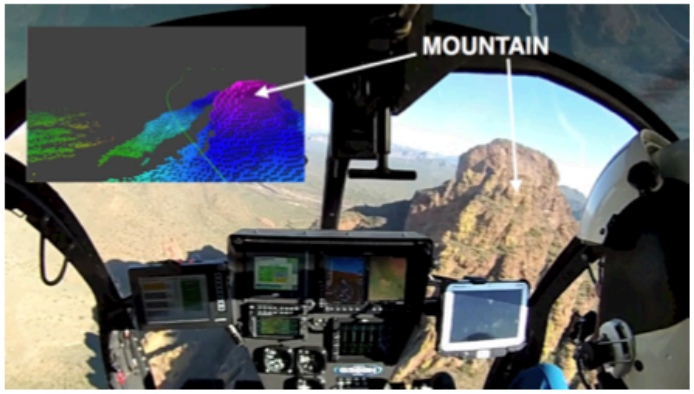
Choudhury and Scherer, ICRA 2015

Outline

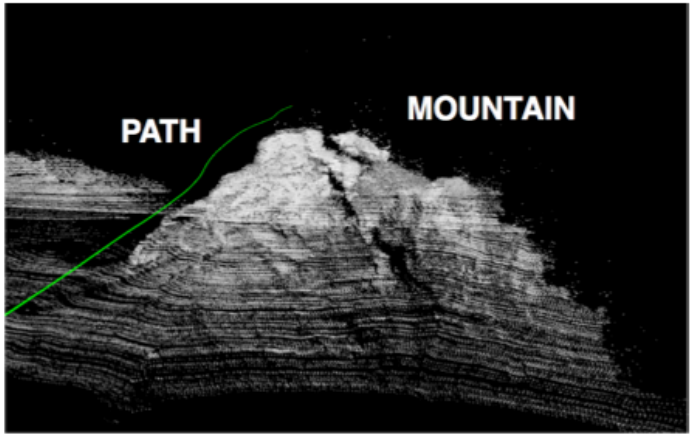
- Introduction
- Problem
- Abstraction and Approach
- **Results**
- Summary
- Exercise

Avoiding a Mountain in the Flight Path

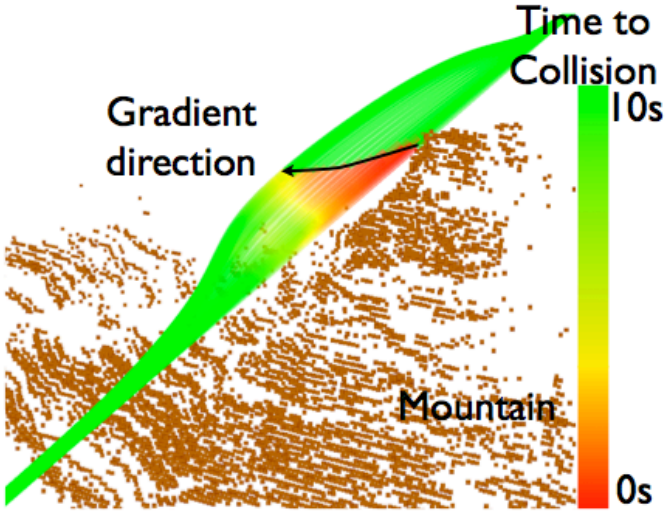
Planning Result for Optimizer.



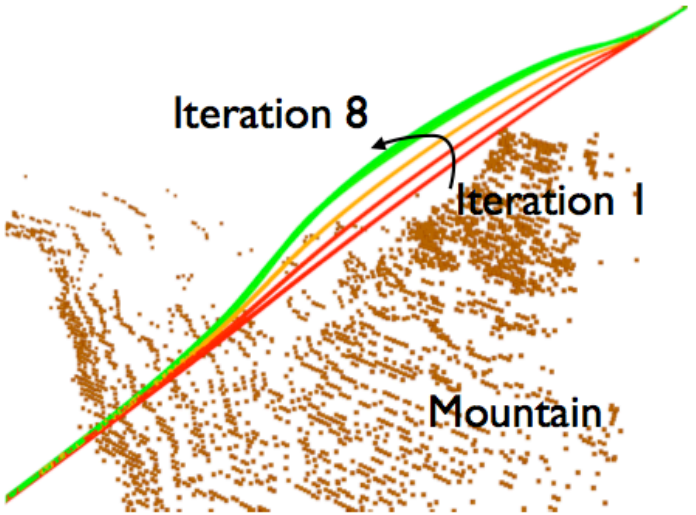
(a)



(b)

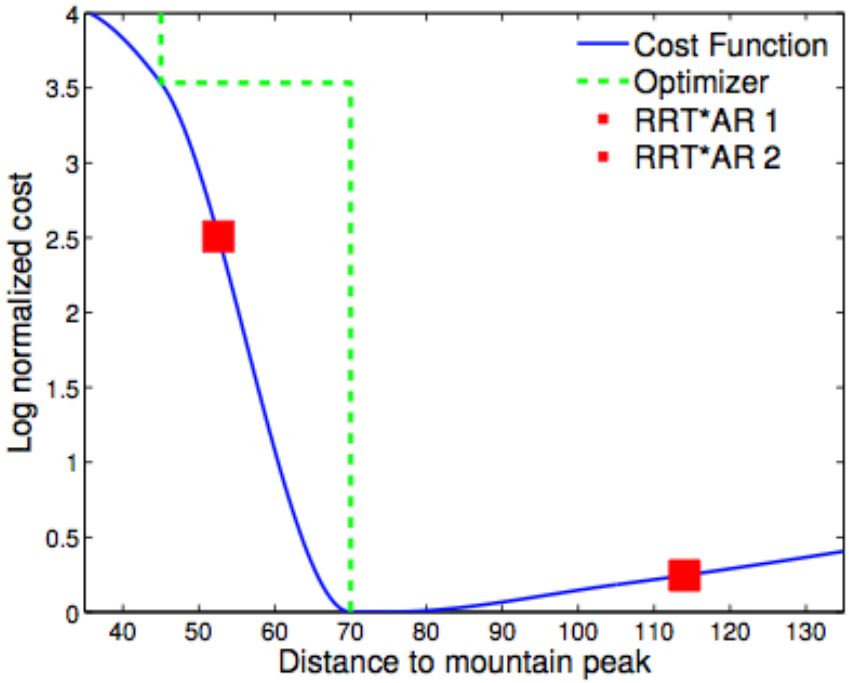


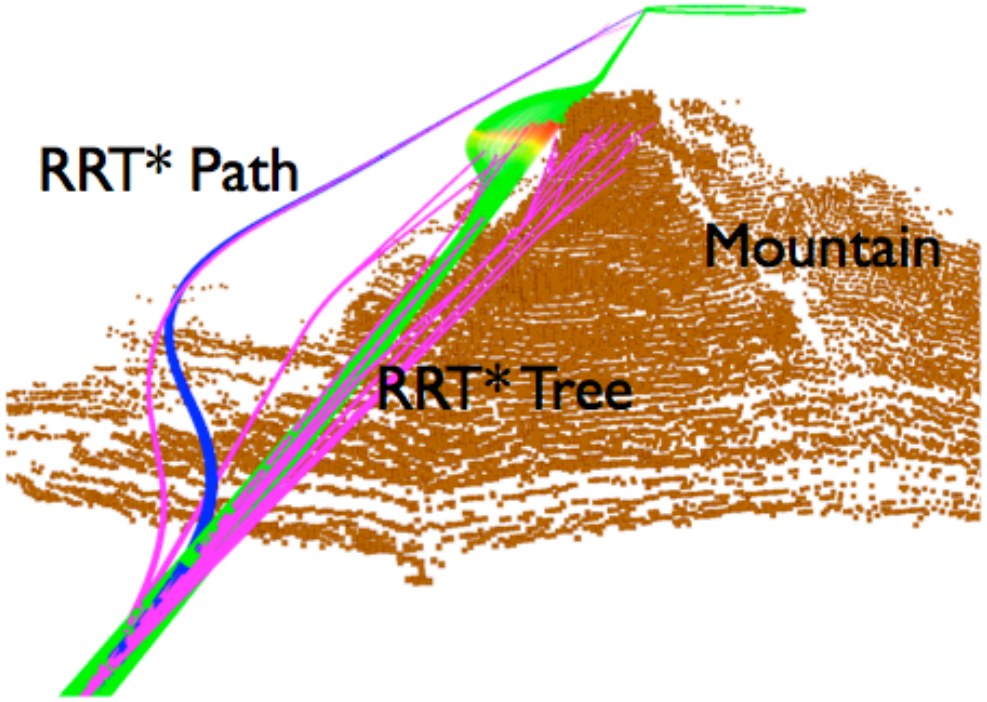
(c)



(d)

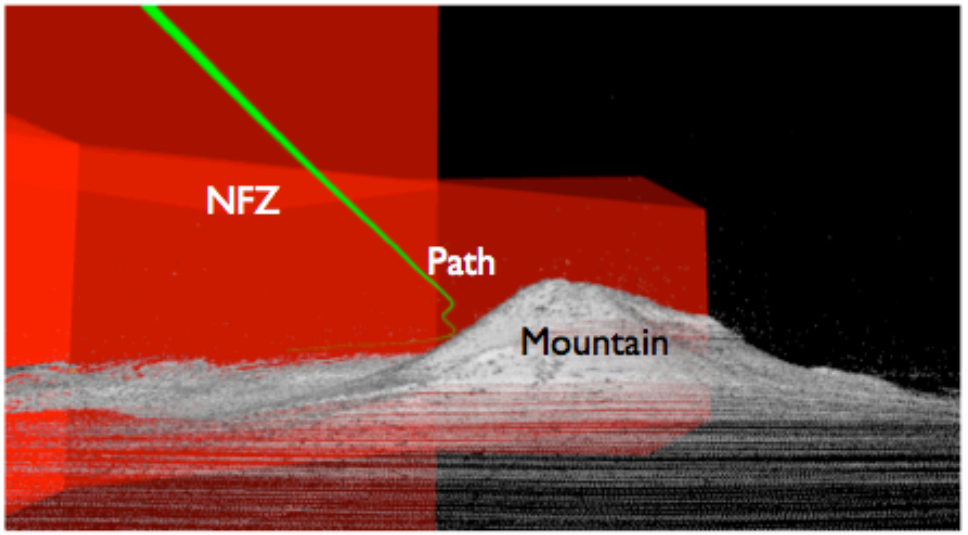
The optimizer can smoothly avoid the mountain and is selected because the cost optimal



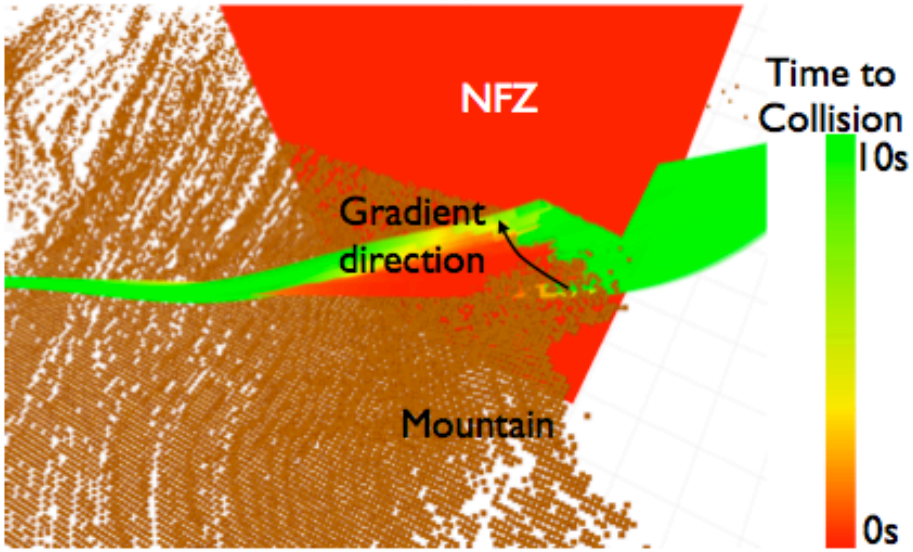


Obstacle Avoidance with a No-Fly-Zone: Comparison between RRT* and Optimizer

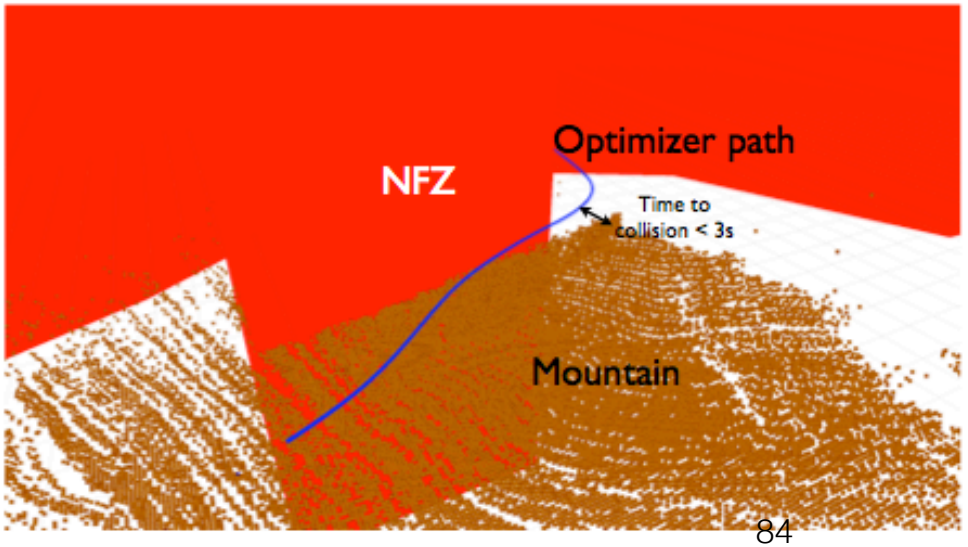
(a)



(b)

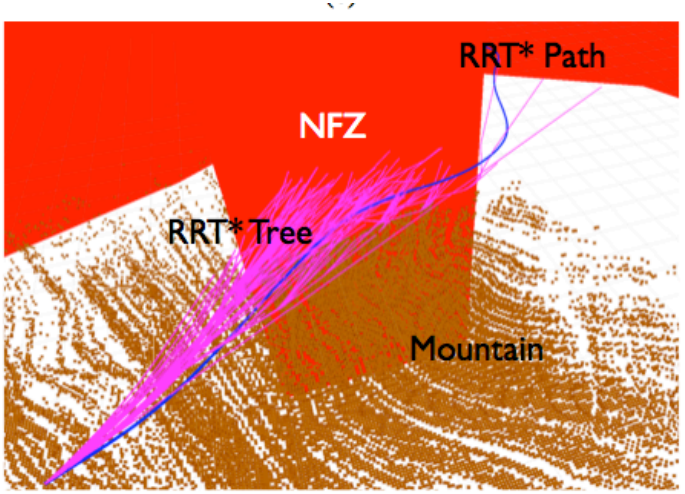


(c)

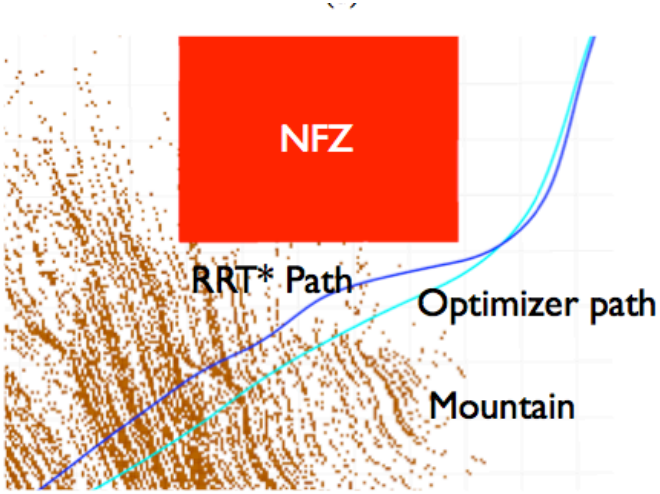


(d)

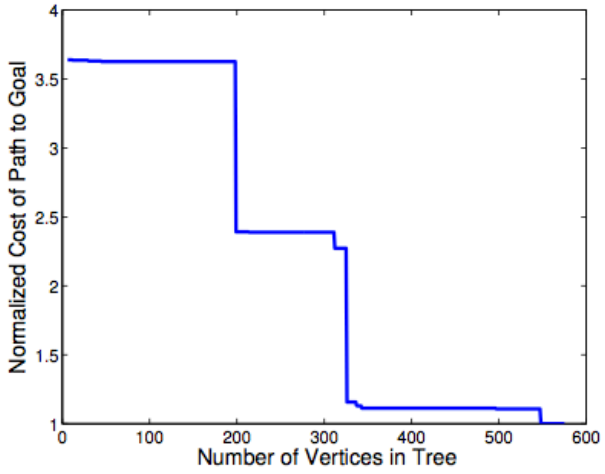
RRT*-AR Path is Picked Because the Optimizer Gets Stuck in a Local Minimum



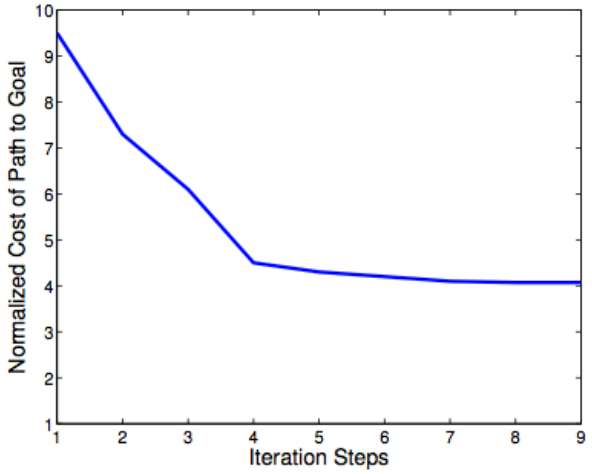
(e)



(f)

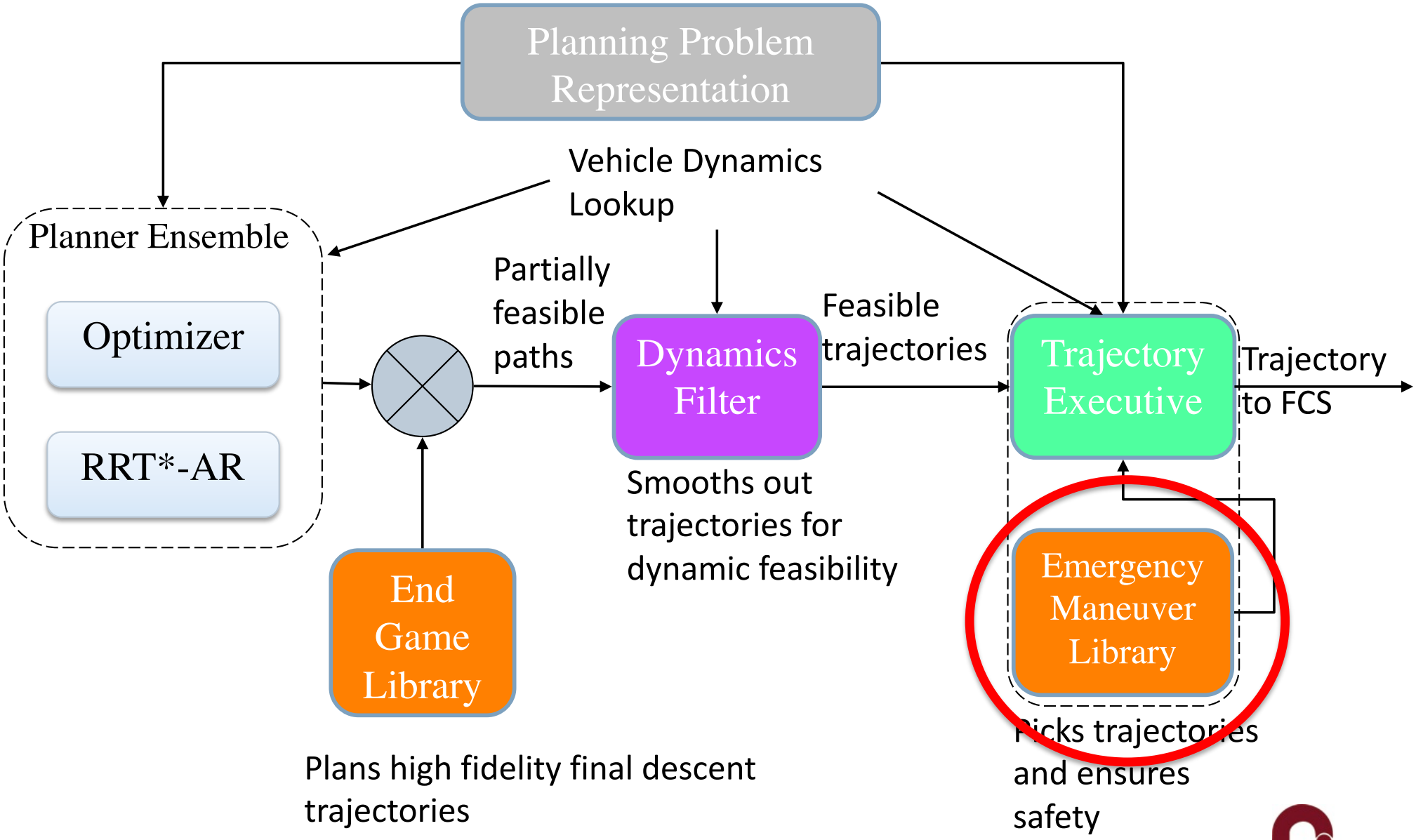


(g)

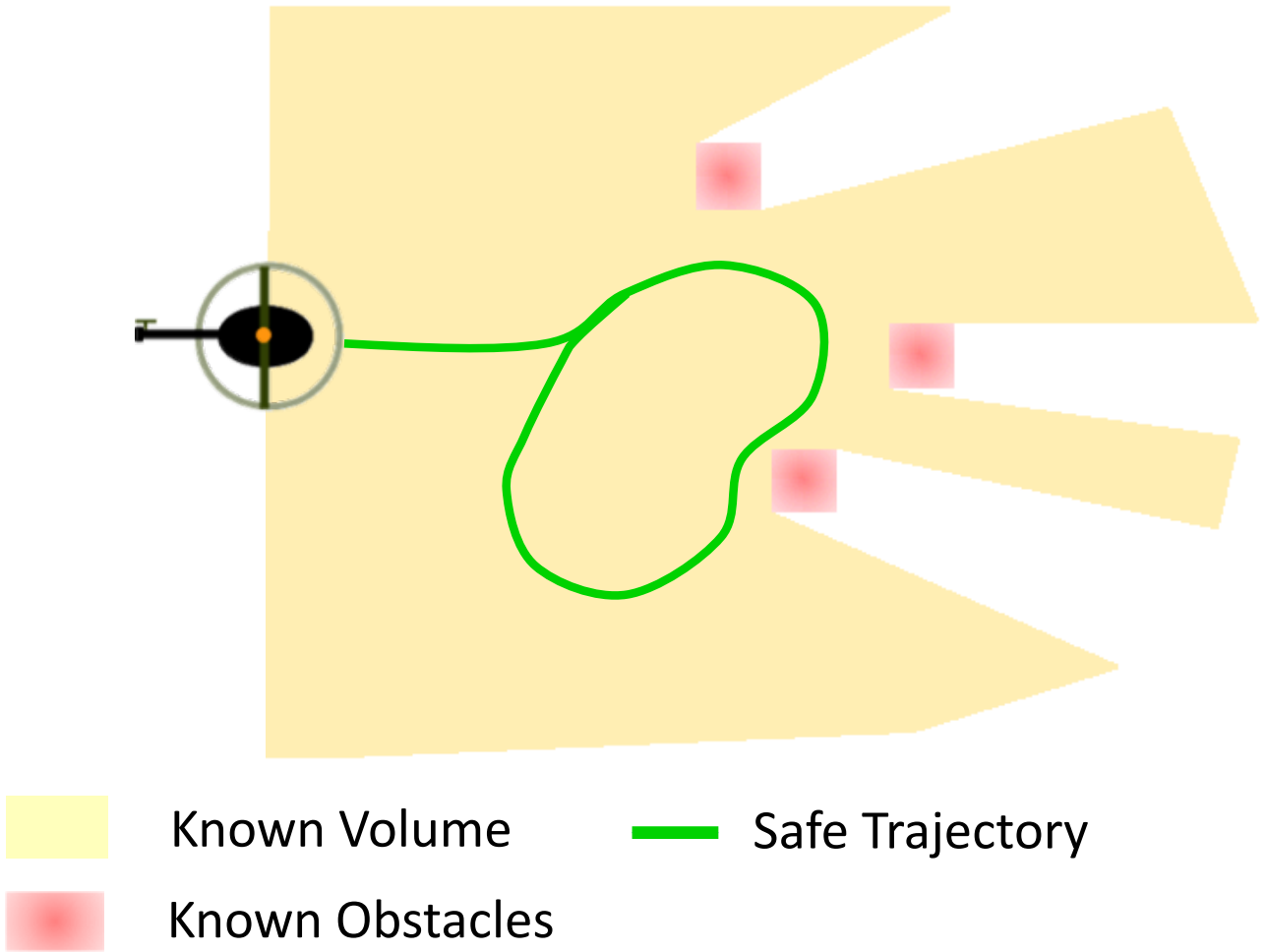


(h)

Trajectory Planner Design



Emergency Maneuver Library – Defining Safety



Emergency Maneuver Problem

$$find \quad \sigma(t) = \{x(t), y(t), z(t), \phi(t), \psi(t), \theta(t)\}$$

$$constraints$$

$$\forall t, \sigma(t) \in K_t / O_t$$ Safety Constraint

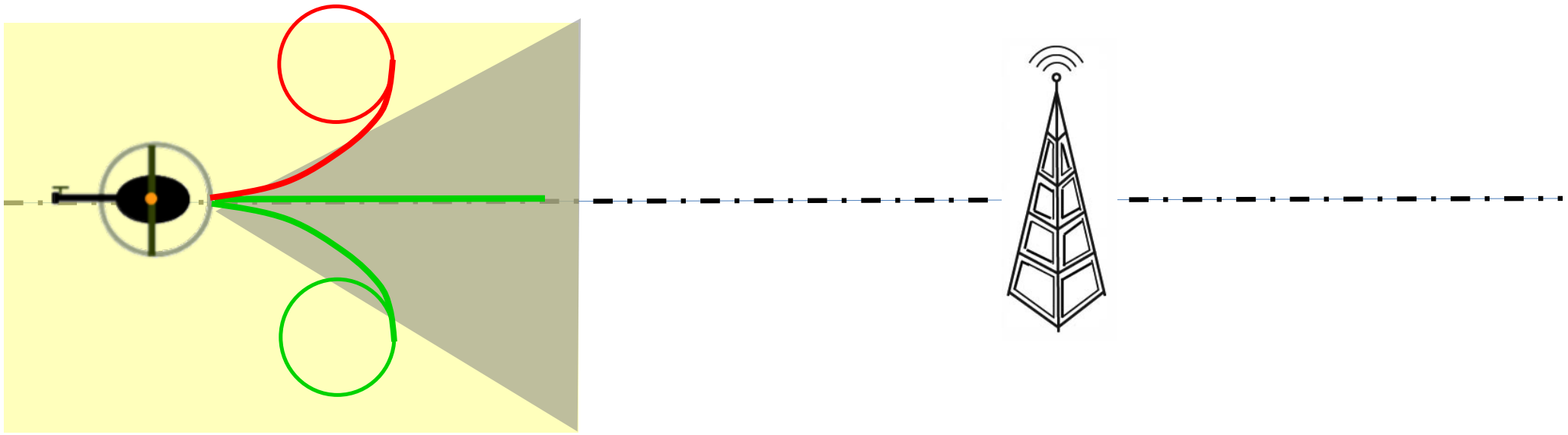
$$\sigma(0) = \sigma_0$$

$$h(\sigma(t), \dot{\sigma}(t), \ddot{\sigma}(t), \ldots) = 0$$

$$g(\sigma(t), \dot{\sigma}(t), \ddot{\sigma}(t), \ldots) \leq 0$$

$\sigma(t)$	Time parameterized trajectory
K_t	Known Volume at time t
O_t	Known Obstacles at time t
$\sigma(0)$	Boundary value constraints
$h(\ldots)$	Non-holonomic equality constraints
$g(\ldots)$	Inequality bounds specifying system limitations

Example Use Case



Planned Trajectory

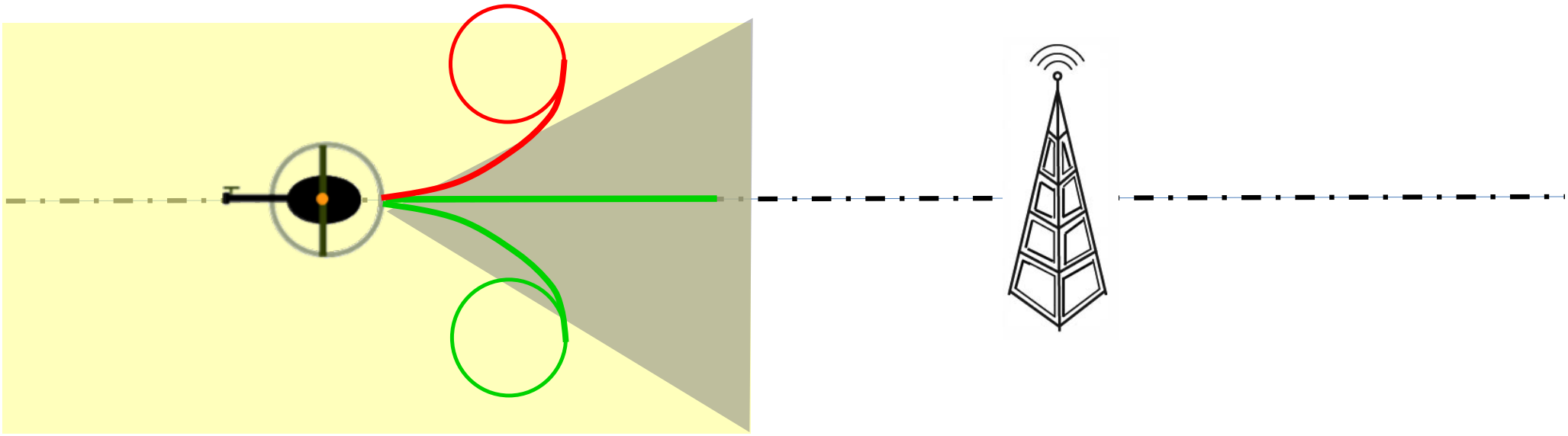
Known Space

Current Laser Coverage

Safe Emergency Maneuver

Unsafe Emergency Maneuver

Example Use Case



Planned Trajectory

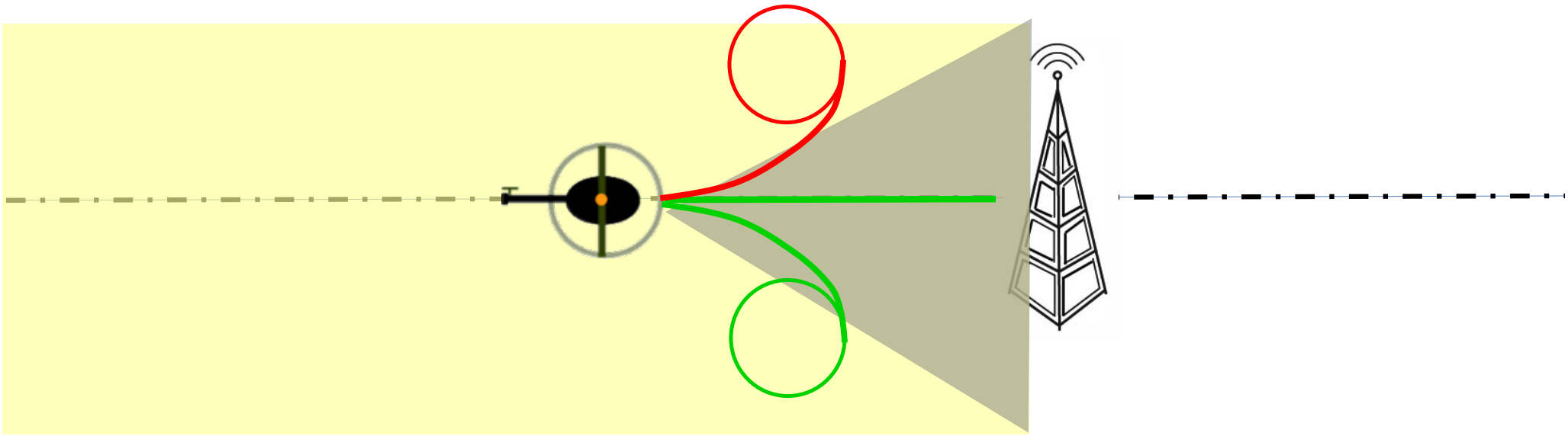
Known Space

Current Laser Coverage

Safe Emergency Maneuver

Unsafe Emergency Maneuver

Example Use Case



Planned Trajectory

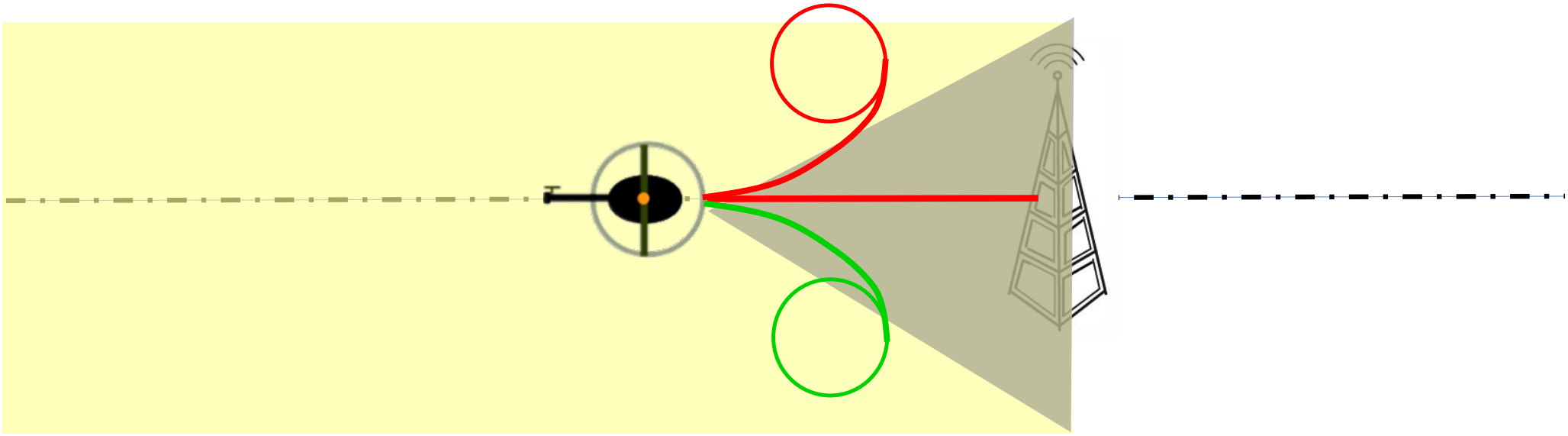
Known Space

Current Laser Coverage

Safe Emergency Maneuver

Unsafe Emergency Maneuver

Example Use Case



Planned Trajectory

Known Space

Current Laser Coverage

Safe Emergency Maneuver

Unsafe Emergency Maneuver

Approach: Emergency Library is Computed Offline to Enable Verification

$$\Upsilon_d = argmax_{\Upsilon} P_u(\Upsilon)$$

$$constraints$$

$$|\Upsilon| \leq n$$

Υ	Set of dynamically feasible control invariant trajectories
n	Number of trajectories allowed in the set
$P_u(.)$	Probability of at least one trajectory in the set being unoccupied

The Emergency Maneuver Set can be Found Greedily

The general optimization problem is NP hard

We greedily generate a set, while proving a sub-optimality bound on the greedy optimization

$$\Upsilon_d = argmax_{\Upsilon} P_u(\Upsilon)$$

$$constraints$$

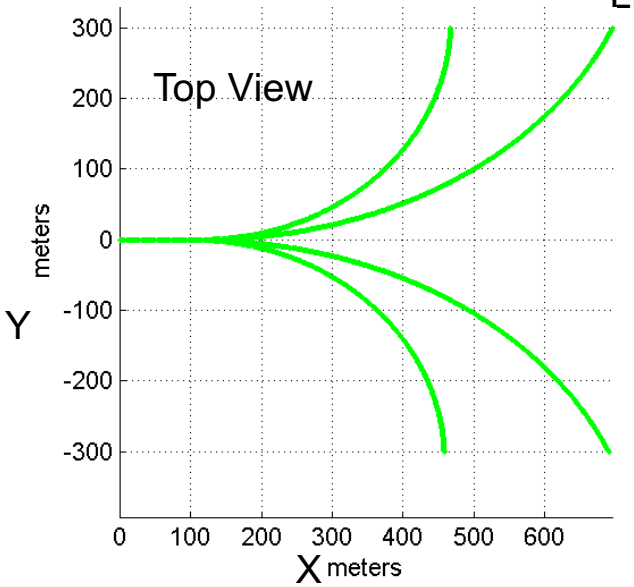
$$|\Upsilon| \leq n$$

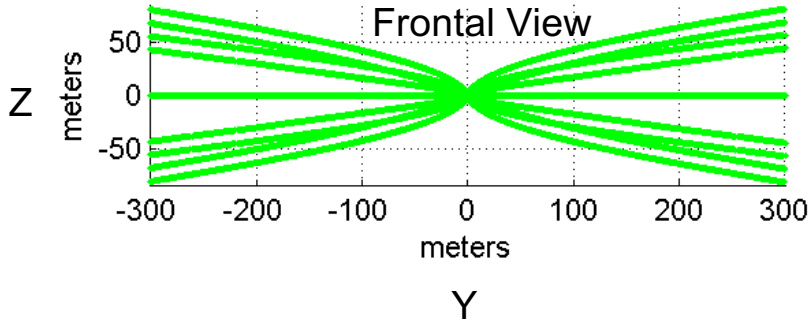
Υ	Set of dynamically feasible control invariant trajectories
n	Number of trajectories allowed in the set
$P_u(.)$	Probability of at least one trajectory in the set being unoccupied

Arora et al., AHS 2014

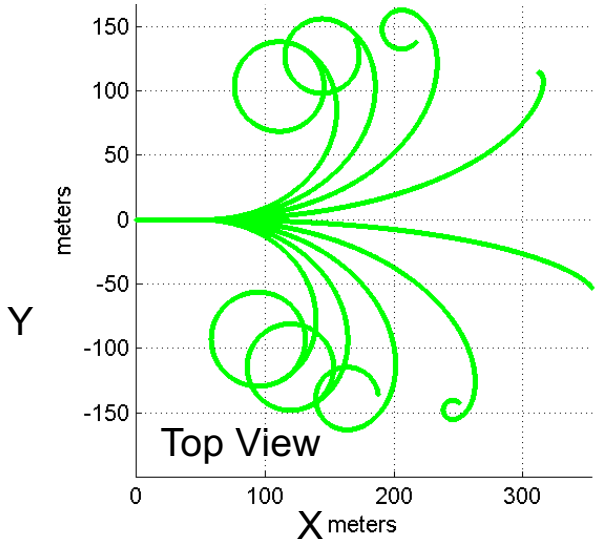
Example Emergency Maneuver Library

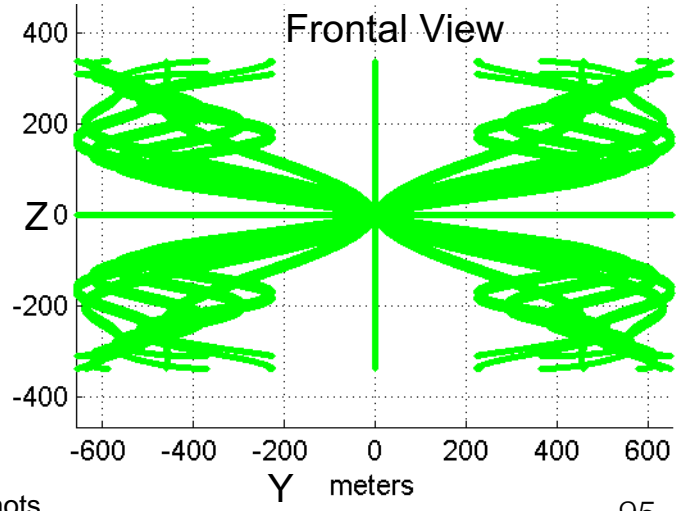
Library at 50 m/s for Swerving (maximum unknown obstacle width 30m)





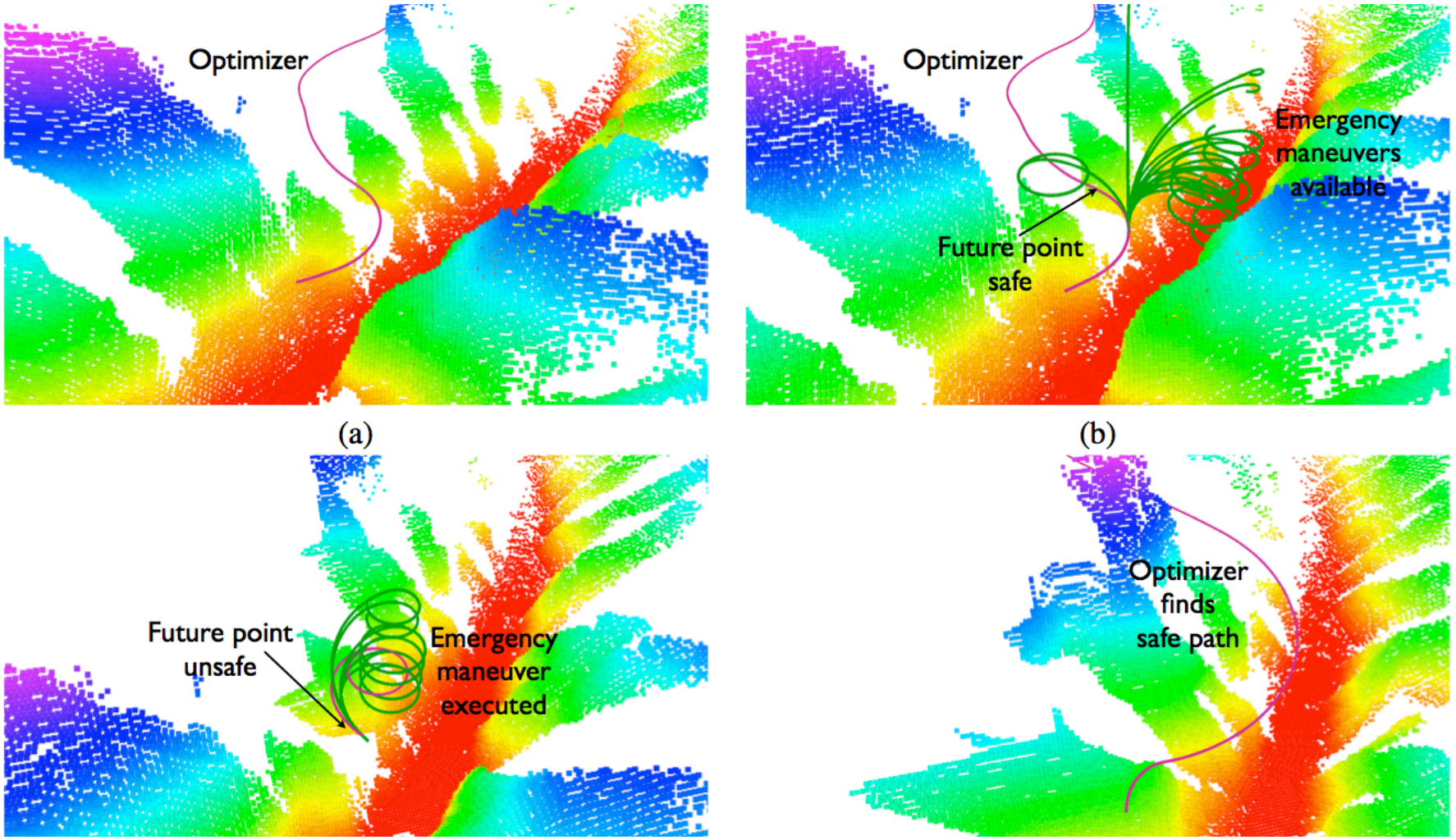
Library at 25 m/s for Stopping



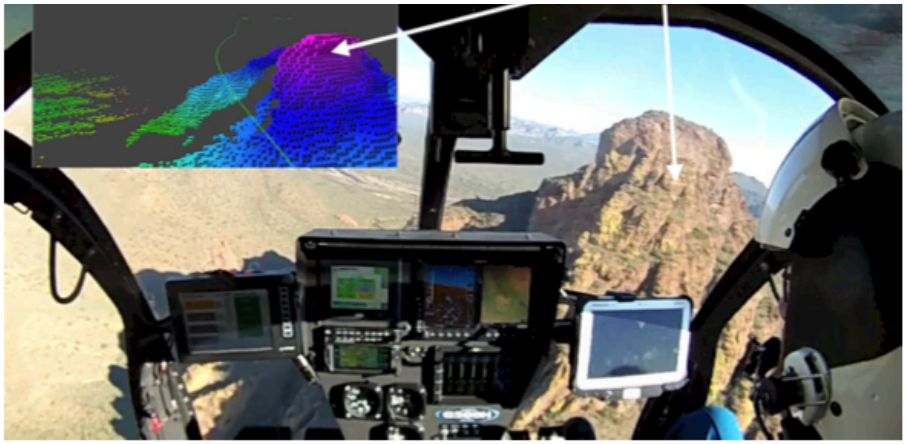


Parameters: wind speed = 0 knots

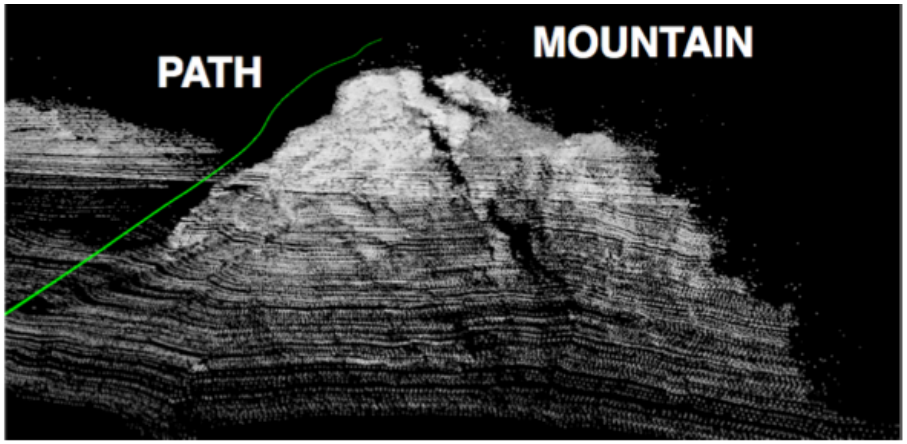
Example Simulation Result



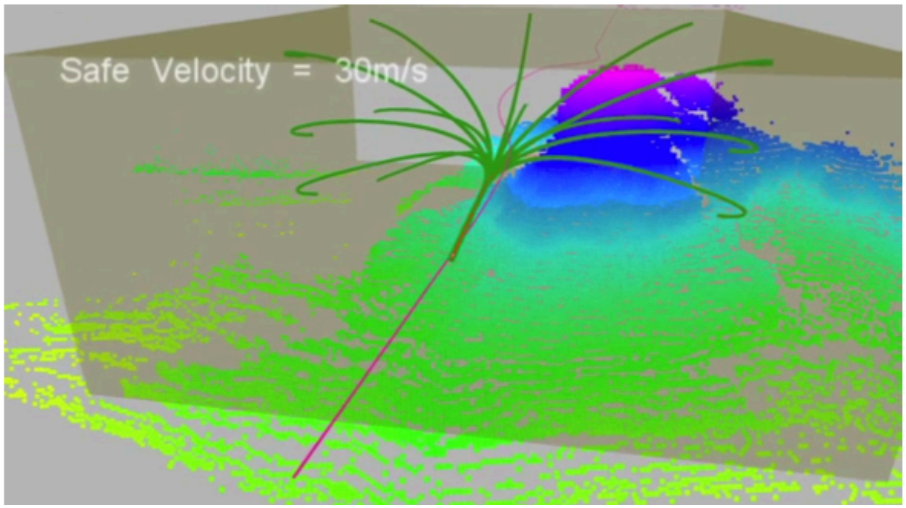
Flight Test Result

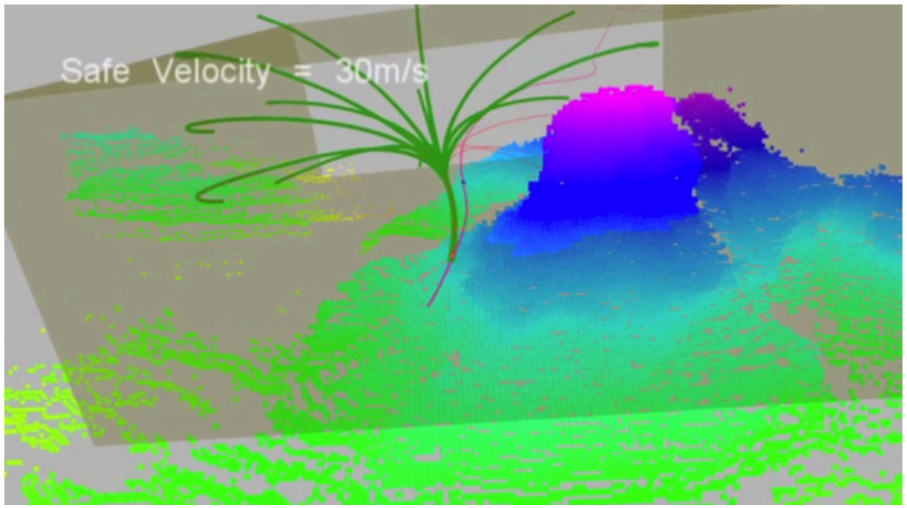


(a)



(b)





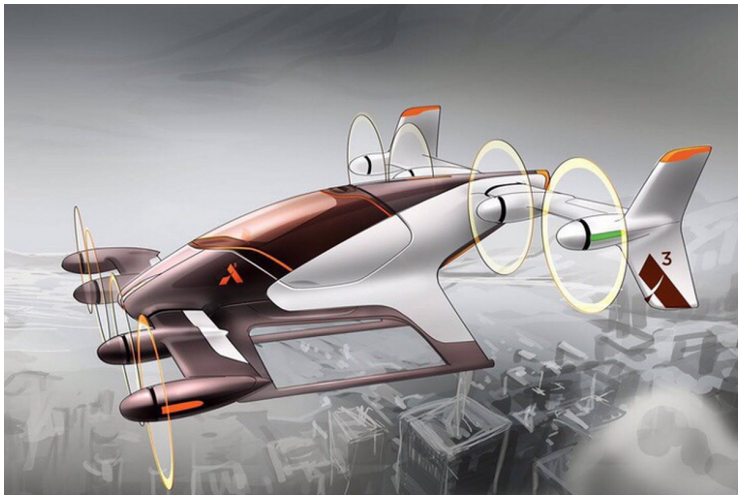
Outline

- Introduction
- Problem
- Abstraction and Approach
- Results
- **Summary & Outlook**
- Exercise

Summary

- Reviewed the planning problem for autonomous flying robots
- Considered several algorithms and the idea of planning ensembles
- Which approaches are successful is highly dependent on your environment and dynamical system

Coming back to the Beginning



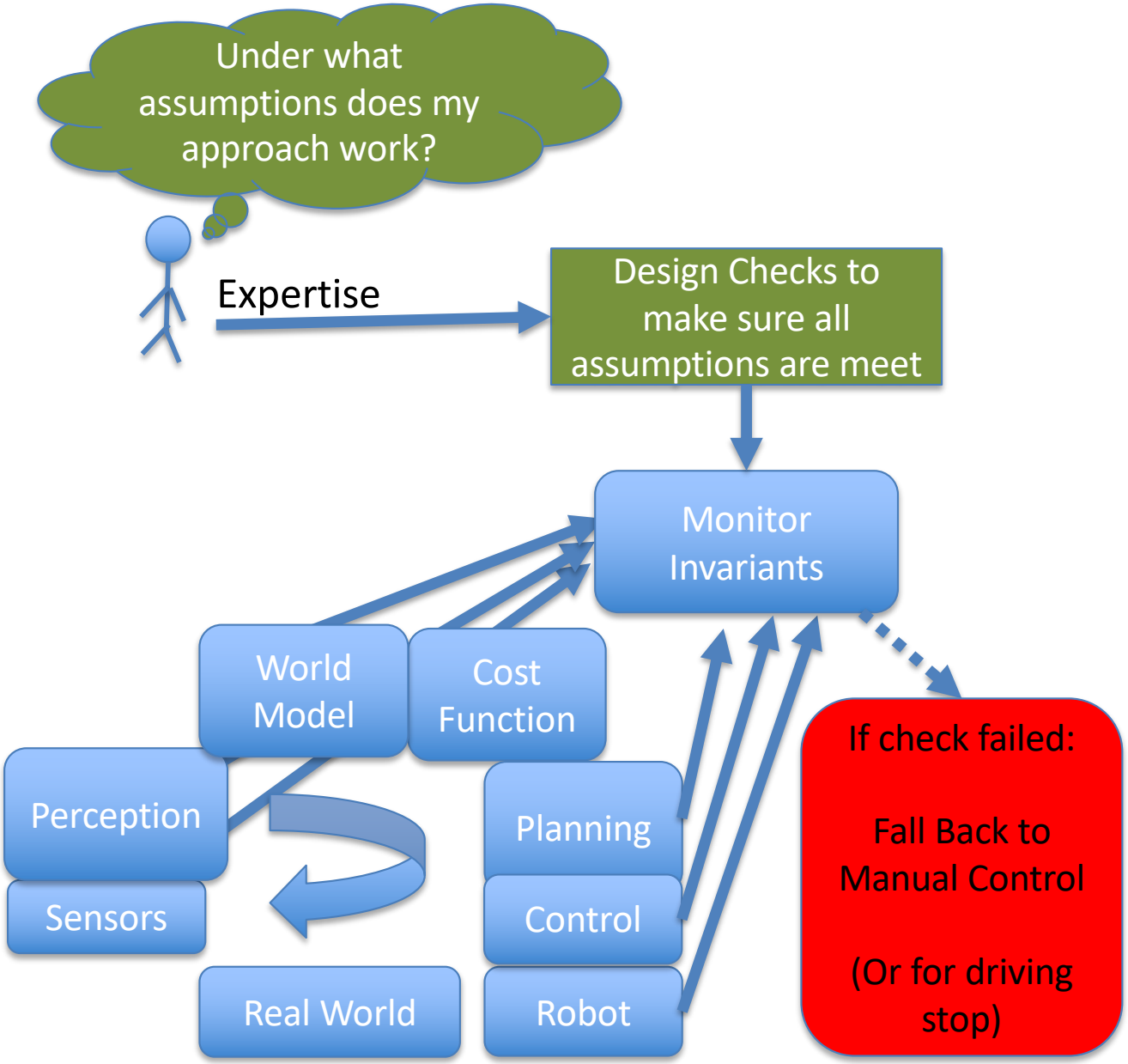
Key Problem: How to react in Off-Nominal Cases?

Emergencies:

- Electrical fire
- GPS-outage
- Bird strike
- Propulsion failure
- Propeller damage
- Door opens in flight
- Unexpected crane in approach
- Person on landing pad

How do you build an autonomy system to correctly react in these situations?

Current Approach (Useful for Self-Driving)



Famous example of problem with this approach:
Air France Flight 447

Report: BEA (France) (July 2012), Final report On the accident on 1st June 2009 to the Airbus A330-203 registered F-GZCP operated by Air France flight AF 447 Rio de Janeiro – Paris https://www.bea.aero/docspa/2009/f-cp090601.en/pdf/f-cp090601.en.pdf

How do we guarantee that we will at least react as good as a pilot?
Paradigm 1: Engineering Paradigm

Case 1: Engine outage in forward flight

Case 2: Engine failure in vertical flight

Case 3: Electrical Fire

Case 4: Passenger Emergency

Case 5: Engine Fire

Case 6: Communications Failure

New Module 1

New Module 1

New Module 1

New Module 1

New Module 1

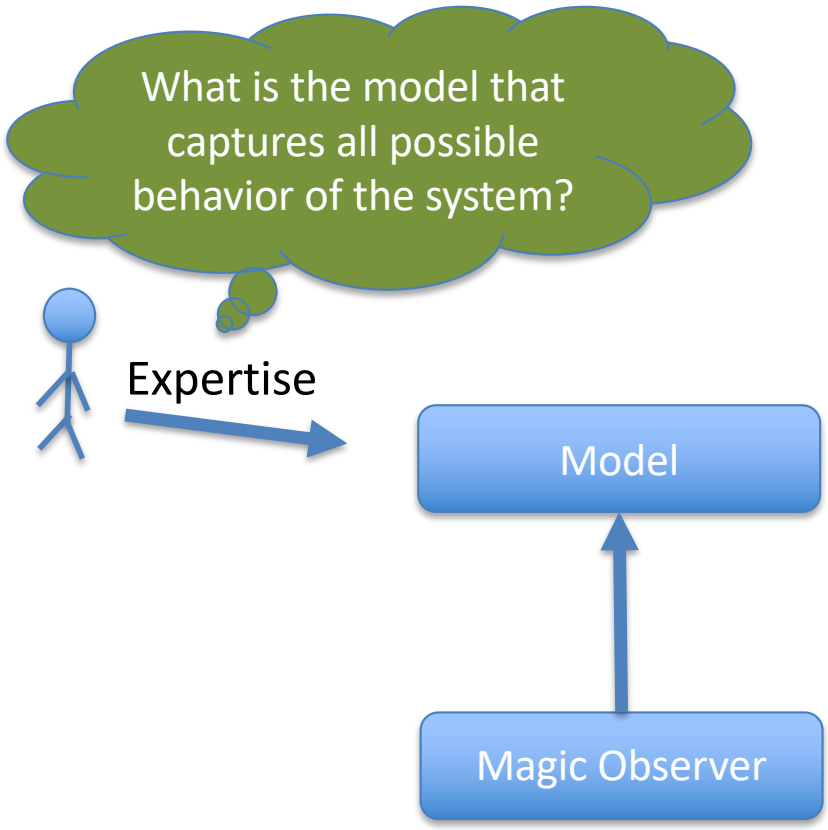
New Module 1

What if 1+4 happen at the same time?

Example Planning for Engine-out Landing

How do we guarantee that we will at least react as good as a pilot?
Paradigm 2: Modelling Approach

You just have overly simplified models if you had the full model you just would have to observe the parameters?

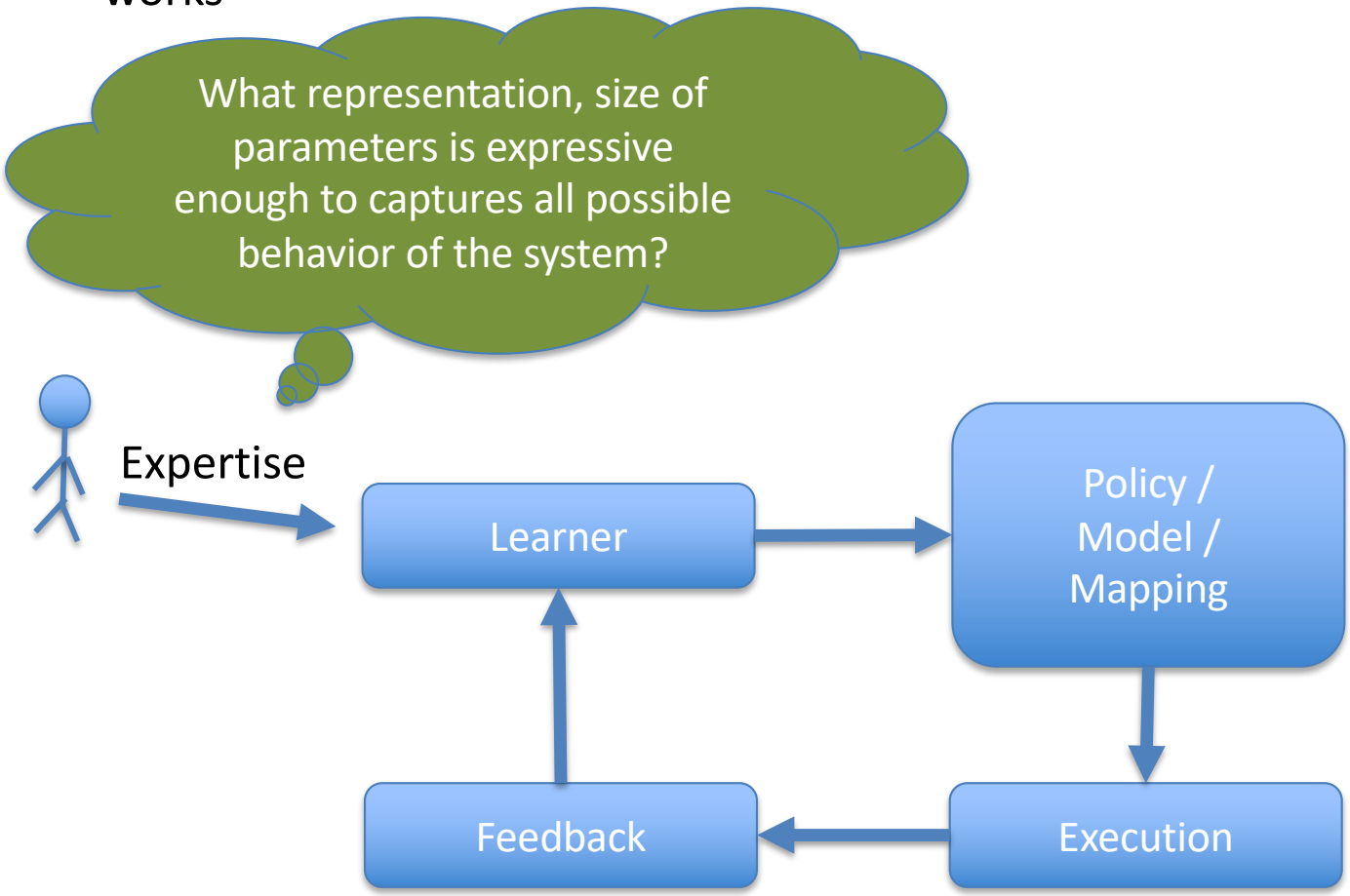


How do you observe all the parameters of the models?

How do we guarantee that we will at least react as good as a pilot?

Paradigm 3: Learning Approach (online)

As long as your learning system is expressiveness you can learn what works



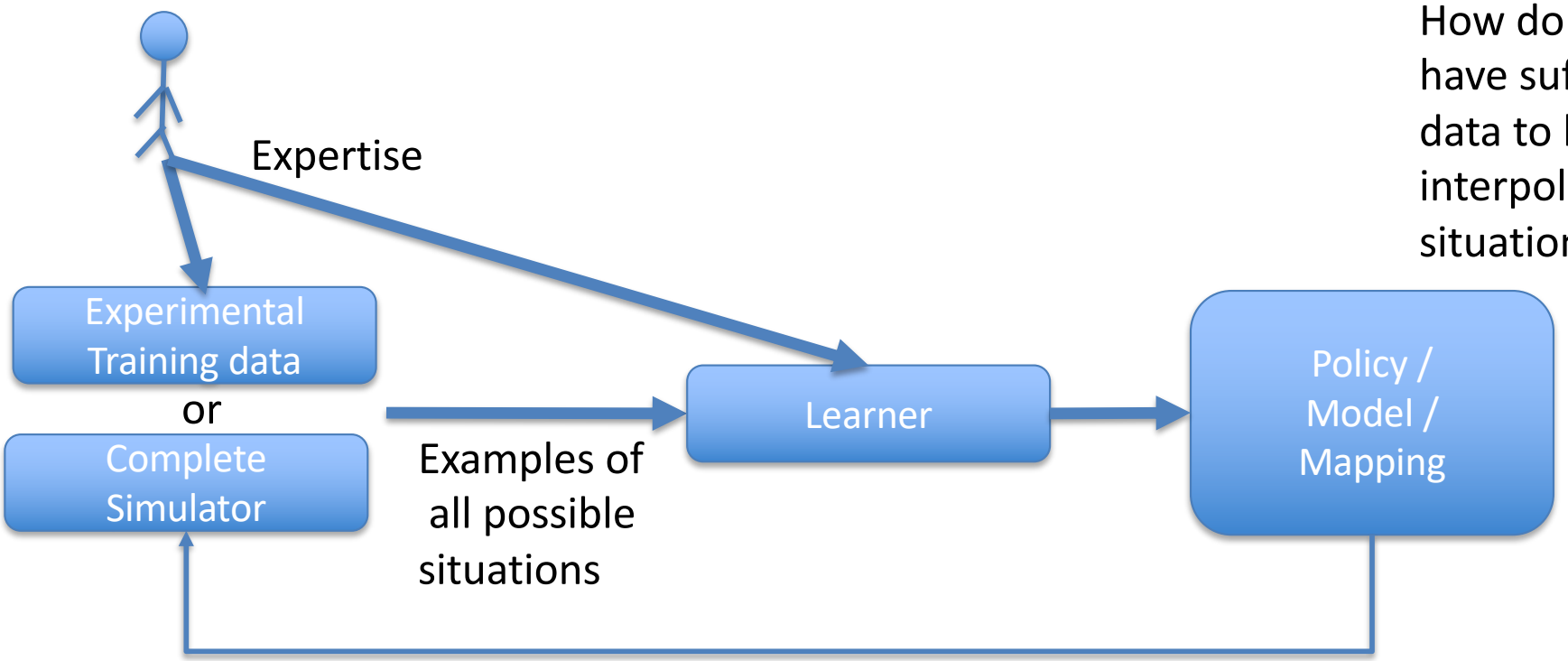
How do you know your learner captures all behaviors?

Will the learning rate be fast enough?

How do we guarantee that we will at least react as good as a pilot?

Paradigm 4: Learning Approach (offline)

As long as your learning system is expressiveness you can learn what works

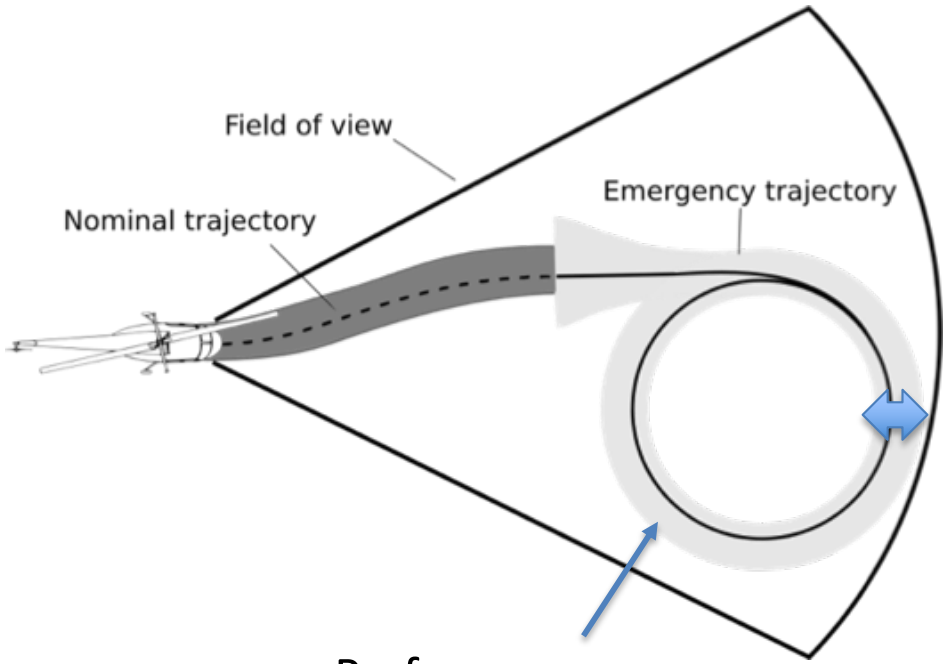


How do you know have sufficient data to be able to interpolate all situations?

Challenge for Formal Guarantees in Autonomy: How tight can one go?

Abstracted models, and requirements / conditions can lead to over approximations

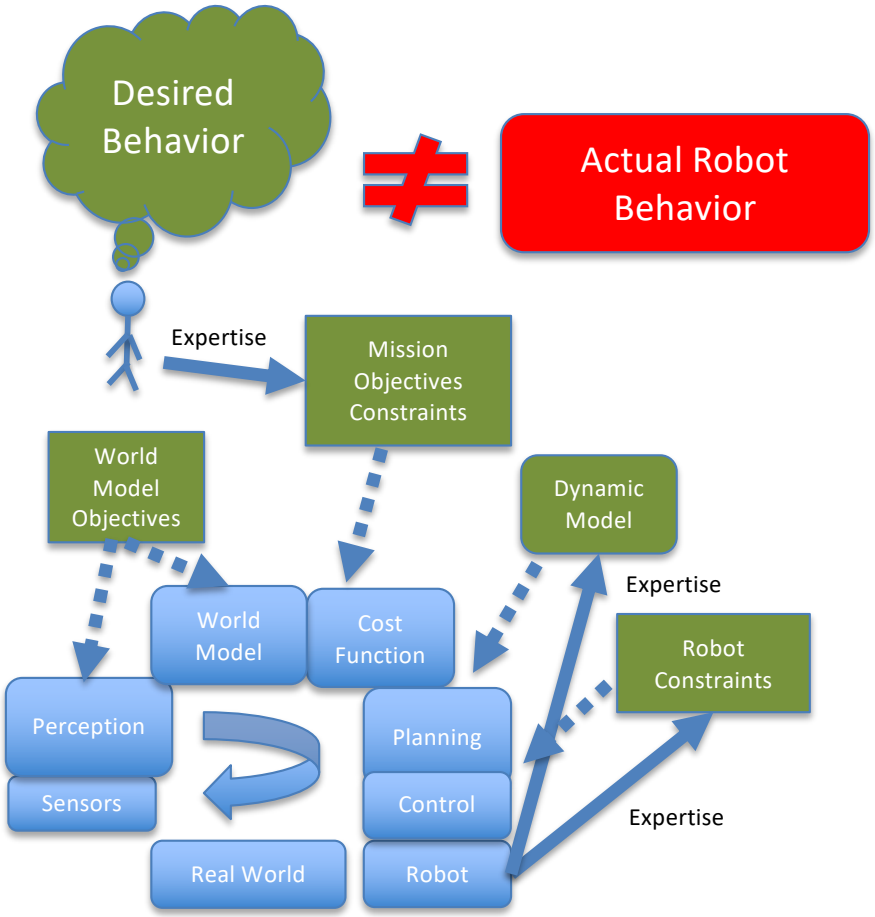
Need tight bounds for sufficient performance

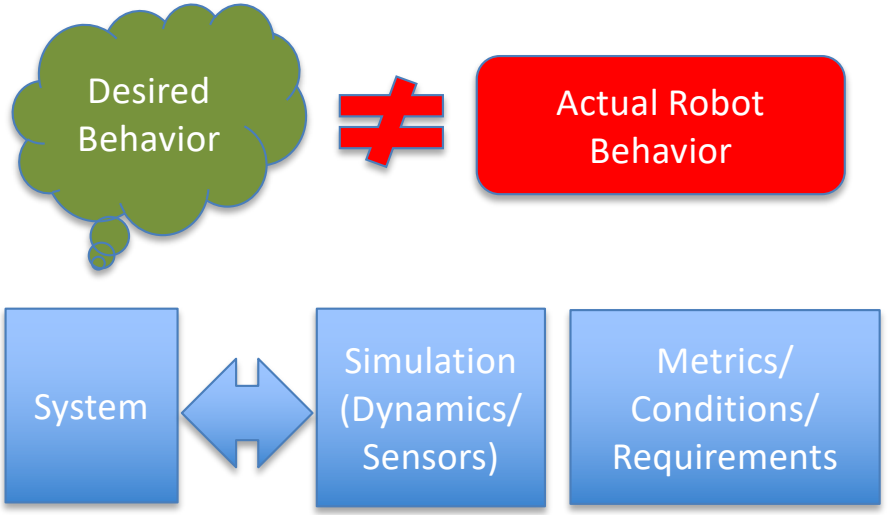


Performance is directly related to size of the reachable set

Challenges: Coverage and Communicating Requirements, and Gaining Trust

How do you define safety conditions efficiently and completely defining the intent of the designer?





Whole system?

Simplified model?

Derived by the designer?

Components?

As realistic as possible?

Given by the simulation?

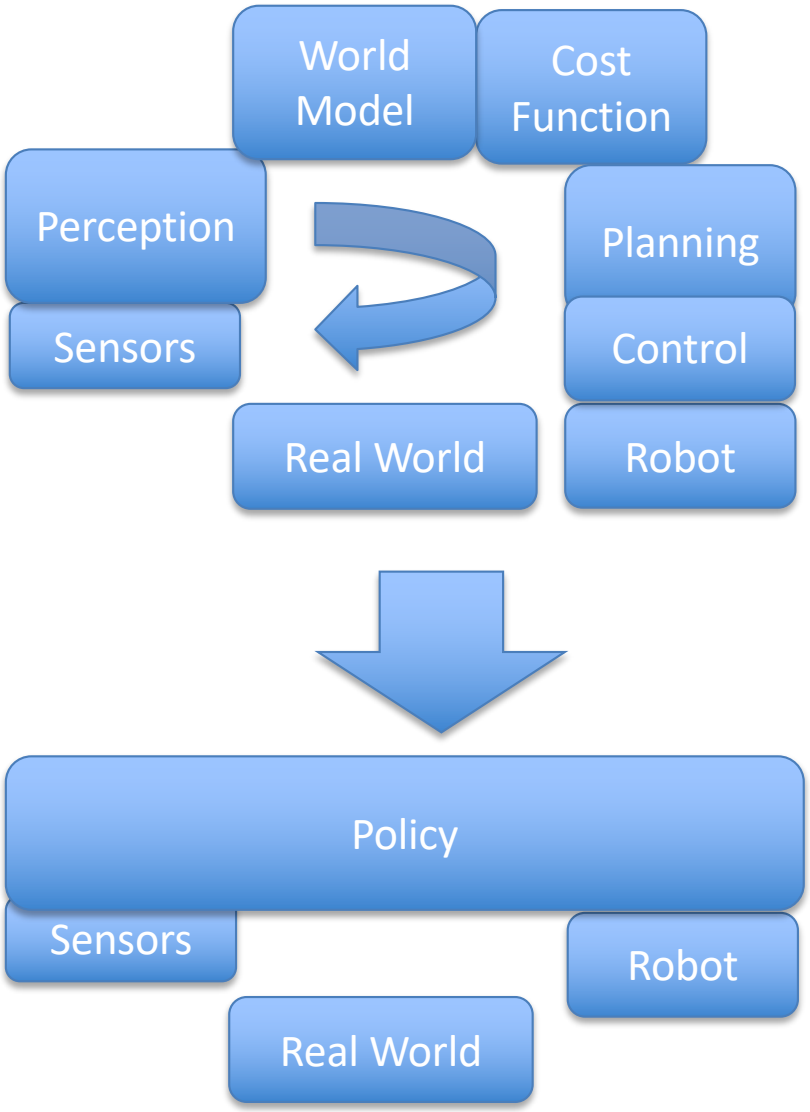
Evaluation or simulation as equivalent to specify formal conditions? (Envelope expansion

Coverage in terms of covering the designers intend?

Challenges: Adaptive Systems?

Bounding the impact of adaptation?

Adaptation is necessary to go beyond the designers intent or effectively capture the intent in changing situations.



Toolbox Setup

1. Install MATLAB (no toolboxes necessary)
2. Download MATLAB toolbox. In a command line execute:

```
git clone
https://bitbucket.org/castacks/matlab_planning_toolbox.git
```

(Supported platforms: Mac, Linux, and
Windows, for Windows you also need Visual C++)

3. Go to
https://bitbucket.org/castacks/matlab_planning_toolbox
and follow the directions.

Outline

- Introduction
- Problem
- Abstraction and Approach
- Results
- Summary
- **Exercise**

Goals

1. Setup the toolbox (10 min)
2. Get familiar with the different planning algorithms from the lecture (20 min)
3. Explore the concept of environment dependence for planning algorithms (20 min)
4. Planning Challenge (40 min)

1. Toolbox Setup

1. Install MATLAB
2. Download MATLAB toolbox. In a command line execute:

```
git clone git@bitbucket.org:castacks/matlab_planning_toolbox.git
```

(Supported platforms: Mac, Linux, and
Windows, for Windows you also need Visual C++)

3. Go to
https://bitbucket.org/castacks/matlab_planning_toolbox
and follow the directions.

1. Matlab Toolbox Overview

- Run `init_setup.m` to setup paths and compile mex files
- Folders
  - `global_search`: Sampling and grid planners
  - `local_search`: Trajectory optimization
  - `cost_functions`: Tools to setup and evaluate cost functions
  - `environment_generation`: Tools to generate environments
  - `saved_environments`: Different maps

2. Get familiar with the different planning algorithms

Different examples are located here:

`planning_experiments/detailed_examples/`

1. `example_astar.m`
2. `example_chomp.m`
3. `example_rrtstar.m`

2. Questions

- How do the different algorithms behave?
- What happens if you vary the parameters?
- What happens if you inflate the heuristic? (Turn A* into Weighted-A*)
- How do you turn RRT* into RRT?

3. Run Algorithms on this Matrix

	CHOMP	A*	RRT*
Env 1			
Env 2			
Env 3			

Where do the algorithms work?

Fill out the matrix.

3. Assumptions so Far:

- No dynamics
- Start
- Goal
- **No changing environment**
- **No time budget/real-time**

4. Planning Challenge

Use this file:
`planning_experiments/planning_challenge/run_planner_challenge.m`

Prize: **Fame and first pick at desert during lunch!**

You can use at most 2 planning algorithms

- Develop your own planning approach on the training set.

- The test set will be released in the last ten minutes. Please report your score in this google doc: https://goo.gl/7O0tBc

References 1

_Rywoi 59a Wxitlir X2Rywoi 0Wermfher Gl syhl yv}0W'~epNemn0Erhvi{ H2Gl eq fiww0Pyoi]shiv0W'fewxmer Wgl iviv0 P}pi N2Gl eq fivpemn0Lykl Gszi v0erh Wermz Wmnkl 0&Eyxsrsq sywl |tpsvexmsr erh Qsxmsr Tpermnk jsver Yrqerrih Eivmep Zil mgpi Rezmkexmnk Vmzi vw0&Nsyvrepsj Jmip h Vsf sxmgw0Nyri 064592

_Evsve59ea Wer oept Evsve0Wermfher Gl syhl yv}0HermipEpl sm0erh W'f ewxmer Wgl iviv0&l q ivkirg} Qeriyzi vPmhvev} Ul rwyvmnk Weji Rezmkexmsr mn Tevxmep} Ors{ r I rzmsrq irxw0&6459 MII I Mxi vrexmsr epGsrji virgi sr Vsf sxmgwerh Eyxsq exmsr 0Qe}064592

_Evsve59f a Wer oept Evsve erh W'f ewxmer Wgl i vi v0&TEWT>Tspmg} Fewi h Ettvsegl jsv W'rwsv Tperrmnk0&6459 MII I Mxi vrexmsr epGsrji virgi sr Vsf sxmgwerh Eyxsq exmsr 0Qe}064592

_Gl syhl yv}59ea Wermfher Gl syhl yv} erh W'f ewxmer Wgl i vi v0&Xl i H}req mgwTvsmigxmsr Jmpxiv,HTJ-1Vi ep Xmqi Rsrpmni evXvemigxsv} Stxmq mexmsr Ywmnk Tvsmigxmsr St i vexsw0&MII I Mxi vrexmsr epGsrji virgi sr Vsf sxmgwerh Eyxsq exmsr 0Qe}064592

_Gl syhl yv}59f a Wermfher Gl syhl yv}0Wer oept Evsve0erh W'f ewxmer Wgl i vi v0&Xl i Tperrivl rwiq fpi>Qsxmsr Tperrmnk f} I |igyxmnk Hmzi wi Ep ksvmxl q w0&MII I Gsrji vi rgi sr Vsf sxmgwerh Eyxsq exmsr 0Qe}064592

_Qexyver e59a HermipQexyvere erh W'f ewxmer Wgl i vi v0&7H Gsrzspyxmsr epRi yvepRix{ svowjsv Perhmnk ^sri Hixigxmsr jvsq PmHEV0&Mxi vrexmsr epGsrji virgi sr Vsf sxmgwerh Eyxsq exmsr 0Qevgl 064592

_Epl sk59a HermipEpl sm erh W'f ewxmer Wgl i vi v0&Gsrrigxi h Mrzevmerx W'xwjsv Lmkl 1W'iih Qsxmsr Tperrmnk mn Tevxmep}1Ors{ r I rzmsrq irxw0&6459 MII I Mxi vrexmsr epGsrji virgi sr Vsf sxmgwerh Eyxsq exmsr 0Qevgl 064592

_Evsve58a Wer oept Evsve0Wermfher Gl syhl yv}0W'f ewxmer Wgl i vi v0erh HermipEpl sm0&E Tvmrgmtpi h Ett vsegl xs I ref pi Weji erh Lmkl Tivjsvq ergi Qeriyzi wwjsv Eyxsrsq sywVsxsvgvejx0&ELW; 4xl Errye pJsvyq 0Qsrxvi ep0 Uyi fi g0Gerehe0Qe} 64Ü660Qe}064582

_Gl syhl yv}58a Wermfher Gl syhl yv}0Wer oept Evsve0erh W'f ewxmer Wgl i vi v0&Xl i Tperrivl rwiq fpi erh Xvemigxsv} I |igyxmzi>E Lmkl Tivjsvq ergi Qsxmsr Tperrmnk W}wxiq { mxl Kyeverxi ih Weji x}0&ELW; 4xl Errye pJsvyq 0Qsrxvi ep0 Uyi fi g0Gerehe0Qe}064582

_Jsi wwi p46a E2Jsi wwi p0ŬWgiri Qshipmnk jvsq Qsxmsr 1Jvii Vehev W'rwmnk0ŮXl i Vsf sxmgwMrwxmxyxi 0GQY0Tmxxwf yvkl 0 TE064462

_Wgl i vi v4=a W2Wgl i vi v0H2Ji vkywsr 0erh W2Wmnkl 0ŬJ н gmnrxG1wt egi erh gswxjyrgxmsr yt hexi wmn 7H jsvyrq errih eivmep zil mgpi w0Ůtvi wirxih exxl i Vsf sxmgwerh Eyxsq exmsr 0644=2MGVE +4=2MII I Mxi vrexmsr epGsrji virgi sr 0644=0 tt 2648=Ü64982

_Keq q i p59a Keq q i pp0N2H2?Wmnmzewe0W2W2?Fevjssx0X2H2O&Fexgl Mjsvq i h Xvi i w,FMX.->Weq tpmnk1fewi h stxmq ep tperrmnk zme xl i liyvmwxmgep} kyrhi h wievgl sj mq tpmgmxverhsq ki sq ixvmg kvetl w0& *Vsf sxmgwerh Eyxsq exmsr ,MGVE-0 6459 MII I Mxi vrexmsr epGsrji virgi sr* 0zsp20rs20tt 274:;074;806:174 Qe} 6459

References 2

[Koenig02] S. Koenig and M. Likhachev, “D* Lite.” In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), pages 476-483, 2002.

[Knepper06] R. Knepper and A. Kelly, “High Performance State Lattice Planning Using Heuristic Look-Up Tables.” Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, October, 2006, pp. 3375-3380.

[LaValle06] S. LaValle, “Planning Algorithms”. Cambridge: Cambridge University Press. ISBN 0521862051.

[Hansen07] E.A. Hansen and R. Zhou, “Anytime Heuristic Search”. Journal of Aritifical Intelligence Research 28 (2007) pages 267-297.

[Likhachev04] Maxim Likhachev, Geoff Gordon and Sebastian Thrun, "ARA*: Anytime A* with Provable Bounds on Sub-Optimality, " Advances in Neural Information Processing Systems 16 (NIPS), MIT Press, Cambridge, MA, 2004.

[Dubins57] L.E. Dubins. “On Curves of Minimum Length with a Constraint on Average Curvature and with Prescribed Initial and Terminal Positions and Tangents,” American Journal of Mathematics, 79: (1957) pages 497-516, 1957.

[Pivtoraiko09] M. Pivtoraiko, R. A. Knepper, and A. Kelly, “Differentially constrained mobile robot motion planning in state lattices,” *Journal of Field Robotics*, vol. 26, no. 3, pp. 308–333, Mar. 2009.

[Frazzoli02] E. Frazzoli, M. Dahleh, and E. Feron, “Real-time motion planning for agile autonomous vehicles,” *AIAA Journal of Guidance Control and Dynamics*, vol. 25, no. 1, pp. 116–129, 2002.

[He13] Keliang He '13, Elizabeth Martin '13, and Matt Zucker. “Multigrid CHOMP with local smoothing” *Proc. IEEE-RAS Int'l Conf. on Humanoid Robotics*, 2013.

[Kolter09] Z. Kolter, and A. Ng “Task-Space Trajectories via Cubic Spline Optimization,” ICRA 2009

[Fang15a] Zheng Fang, Shichao Yang , Sezal Jain, Geetesh Dubey, Silvio Mano Maeta, Stephan Roth, Sebastian Scherer, Yu Zhang, and Stephen T. Nuske, "Robust Autonomous Flight in Constrained and Visually Degraded Environments," Field and Service Robotics, June, 2015.

[Holtz15] Kristen Holtz, Daniel Maturana, and Sebastian Scherer, "Learning a Context-Dependent Switching Strategy for Robust Visual Odometry," Field and Service Robotics, June, 2015.

[Fang15b] Zheng Fang and Sebastian Scherer, "Real-time Onboard 6DoF Localization of an Indoor MAV in Degraded Visual Environments Using a RGB-D Camera," 2015 IEEE International Conference on Robotics and Automation, May, 2015.

[Choudhury16] S. Choudhury, J. D. Gammell, T. D. Barfoot, S. Srinivasa, and S. Scherer, “Regionally accelerated batch informed trees (RABIT*): A framework to integrate local information into optimal path planning,” presented at the IEEE International Conference on Robotics and Automation ICRA, Stockholm, Sweden, 2016, pp. 4207–4214.