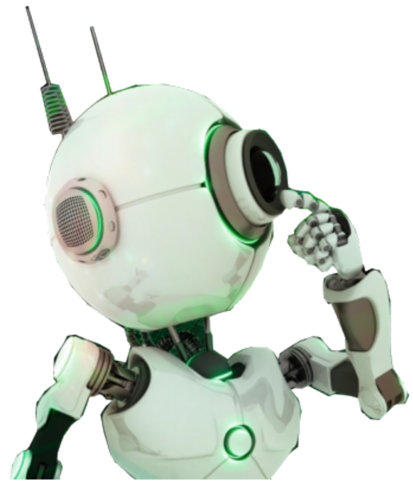
Risk-bounded Planning



Prof. Brian Williams

June 16^{th}, 2017

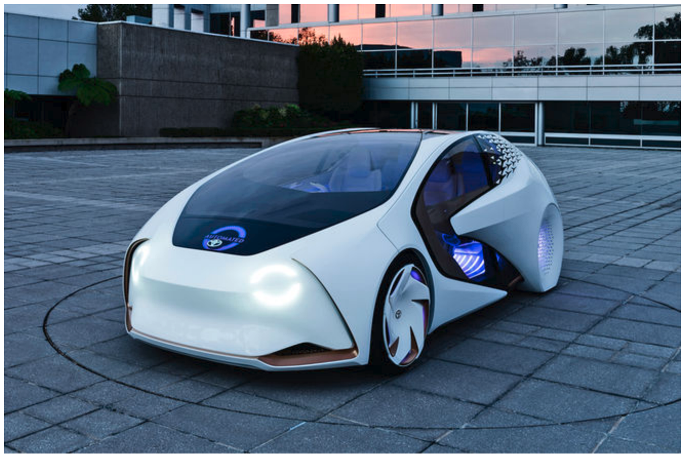
Cognitive Robotics Summer School

photo courtesy MIT News

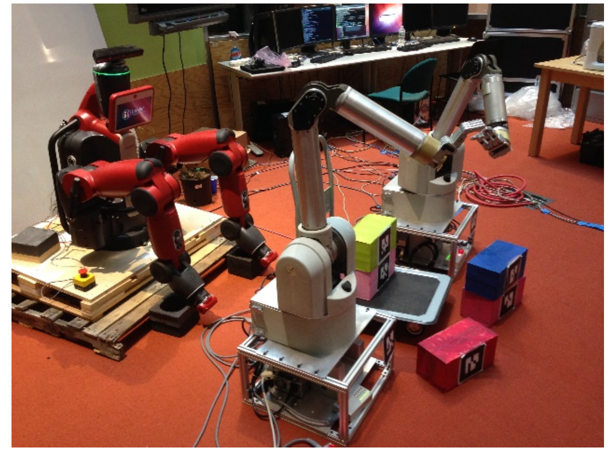
Related Papers

- T. Léauté and B. Williams, “Coordinating Agile Systems Through the Model-based Execution of Temporal Plans," *AAAI-05,* Pittsburgh, PA, July 2005, pp. 114-120.
- A. Hofmann and B. Williams, “Exploiting Spatial and Temporal Flexibility for Plan Execution of Hybrid, Under-actuated Systems," *AAAI-06,* Boston, MA, July 2006, pp. 948-955.
- L. Blackmore, “A Probabilistic Particle Control Approach to Optimal, Robust Predictive Control,” Proceedings of the AIAA Conference on Guidance Navigation and Control, 2006.
- M. Ono and B. C. Williams, "Iterative Risk Allocation: A New Approach to Robust Model Predictive Control with a Joint Chance Constraint,” *IEEE Conference on Decision and Control,* Cancun, Mexico, 2008.
- M. Ono, B. Williams and L. Blackmore, “Probabilistic Planning for Continuous Dynamic Systems under Bounded Risk,” Journal of Artificial Intelligence Research, v. 46, 2013.
- M. Ono, B. Williams, and L. Blackmore, “Probabilistic Planning for Continuous Dynamic Systems under Bounded Risk,” Journal of Artificial Intelligence Research, 2013.
- Vidal, T and Fargier, H., "Handling contingency in temporal constraint networks: from consistency to controllability." *Journal of Exp. & Theo. AI* 11.1 (1999): 23-45.
- C. Fang, P. Yu, P., and B. Williams, “Chance-constrained Probabilistic Simple Temporal Problems.” AAAI 14.
- A. Wang and B. Williams, “Chance-constrained Scheduling via Conflict-directed Risk Allocation.” AAAI 15.
- P. Santana, T. Vaquero, C. Toledo, A. Wang, C. Fang & B. Williams, “PARIS: a Polynomial-Time, Risk-Sensitive Scheduling Algorithm for Probabilistic Simple Temporal Networks with Uncertainty,” ICAPS 16.
- E. Hansen & S. Zilberstein, “LAO*: a heuristic search algorithm that finds solutions with loops”, Artificial Intelligence vol. 129 (1), 35-62, 2001.
- P. Santana, S. Thiebaux and B. Williams, “RAO*: An Algorithm for Chance-Constrained POMDP's,” AAAI 16.
- F. Trevizan, S. Thiebaux, P. Santana, and B. Williams, “Heuristic Search in Dual Space for Constrained Stochastic Shortest Path Problems,” ICAPS 16.

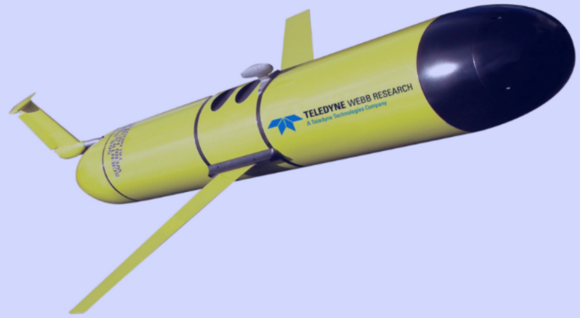
Model-based Embedded
& Robotic Systems group



Autonomous Cars



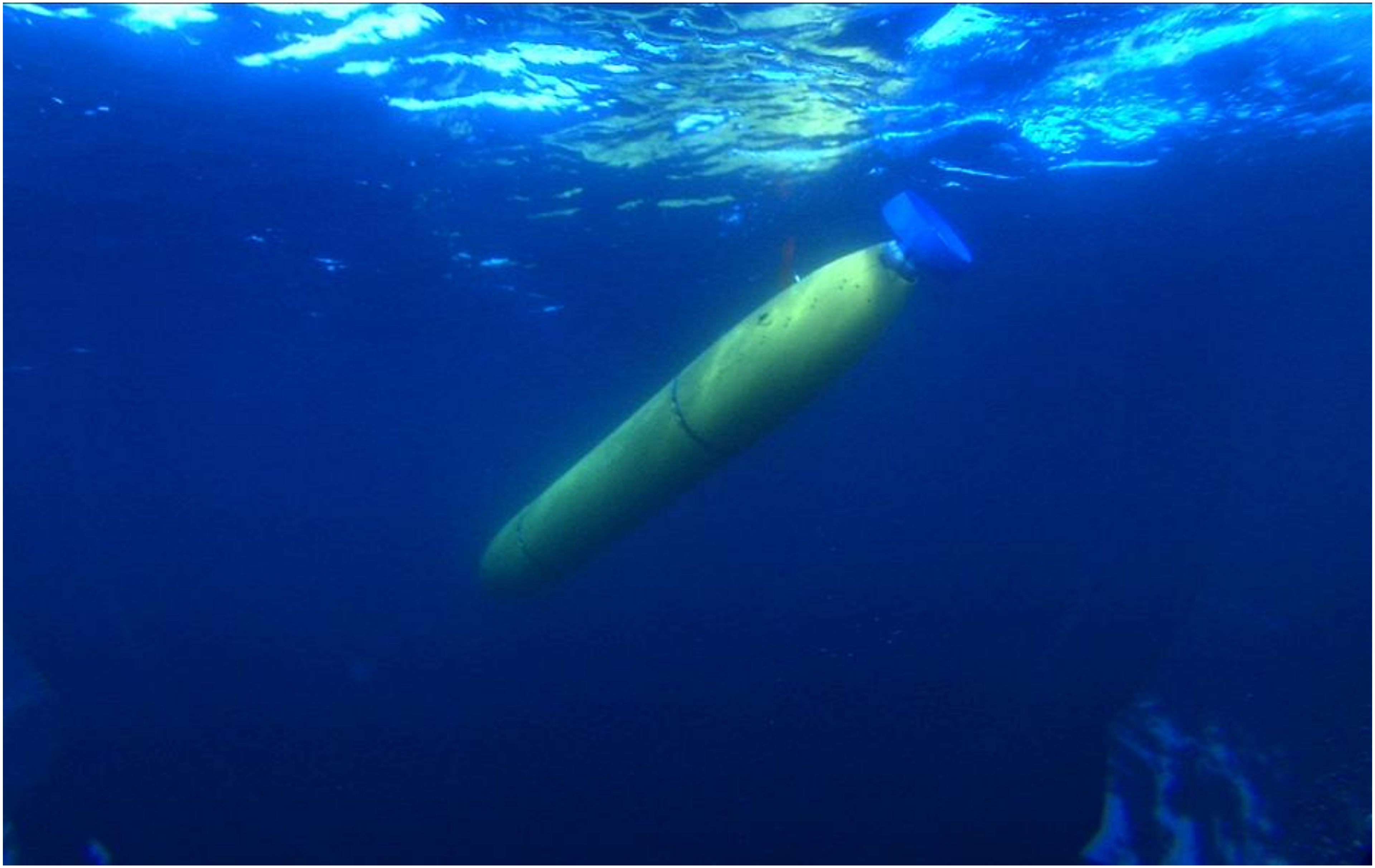
Human-Robot Teams



Information Gathering Scouts

Robots need to be safe → Make robots risk-aware

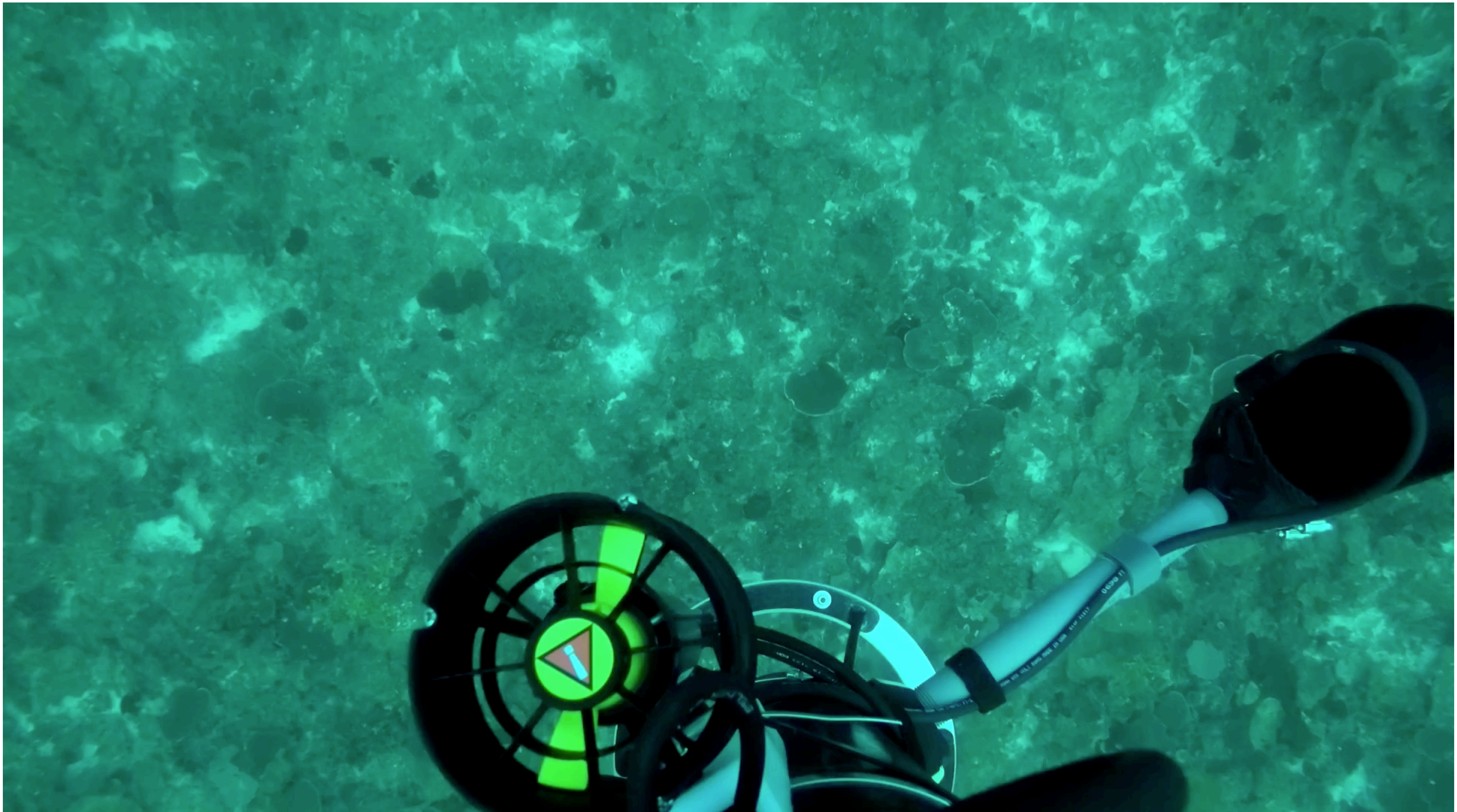
Uncertainty, When Ignored, Can Lead to Failure



Courtesy MBARI

Problem: Coordinating Networks of Vehicles to Perform Time-critical Missions in Risky Environments

Even in a Large Work Area, (Near) Collisions Can Happen



Need to manage uncertainty and risk at multiple levels

"How long does this activity run?"

Temporal uncertainty

"Where is the robot?"

State uncertainty

"Did the instrument turn on or go safe?"

Enumeration uncertainty

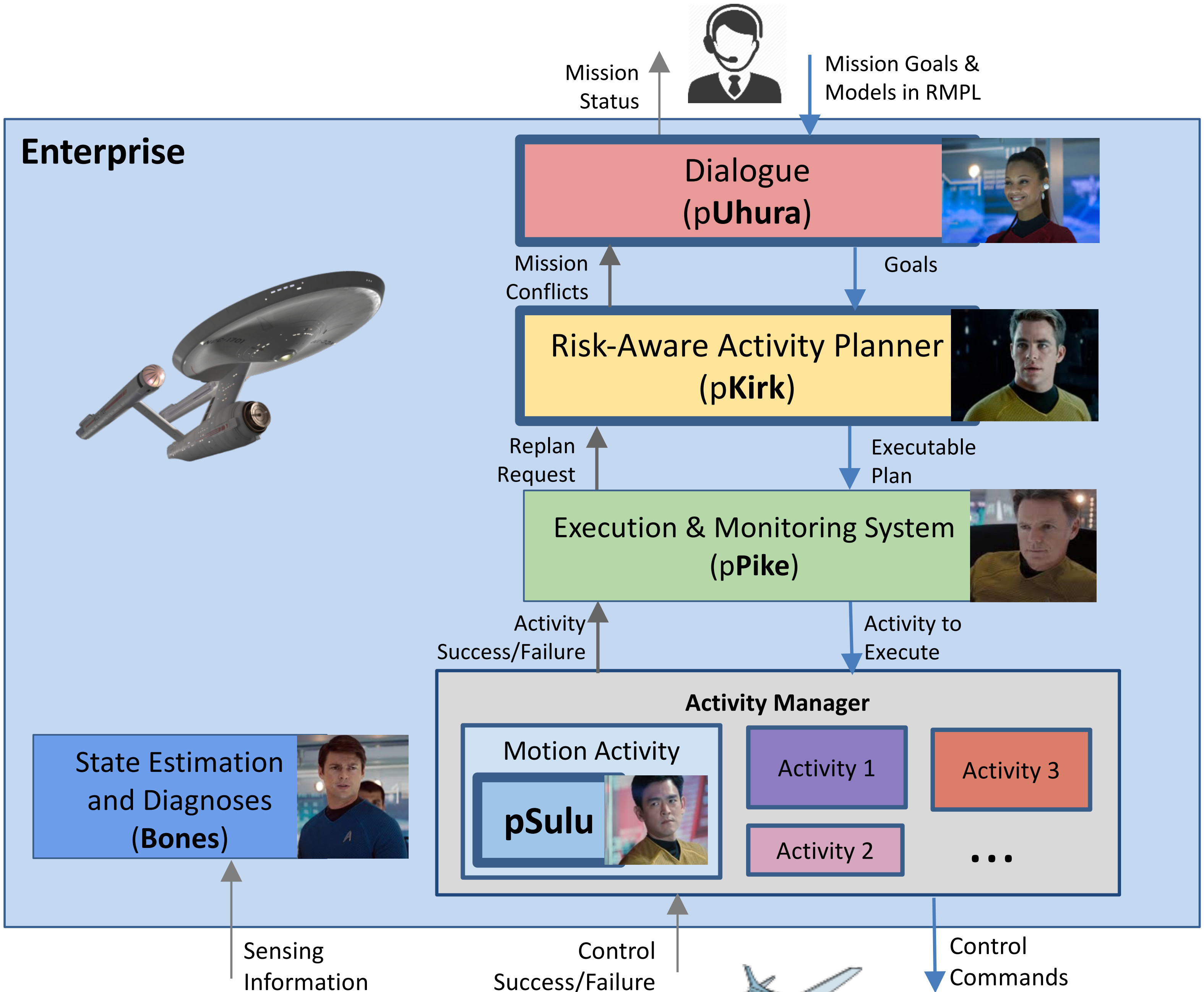
"Where did this Gaussian come from?"

Model uncertainty

"How much charge is left in the battery?"

Resource uncertainty

Enterprise: Risk-aware Goal-directed Executive



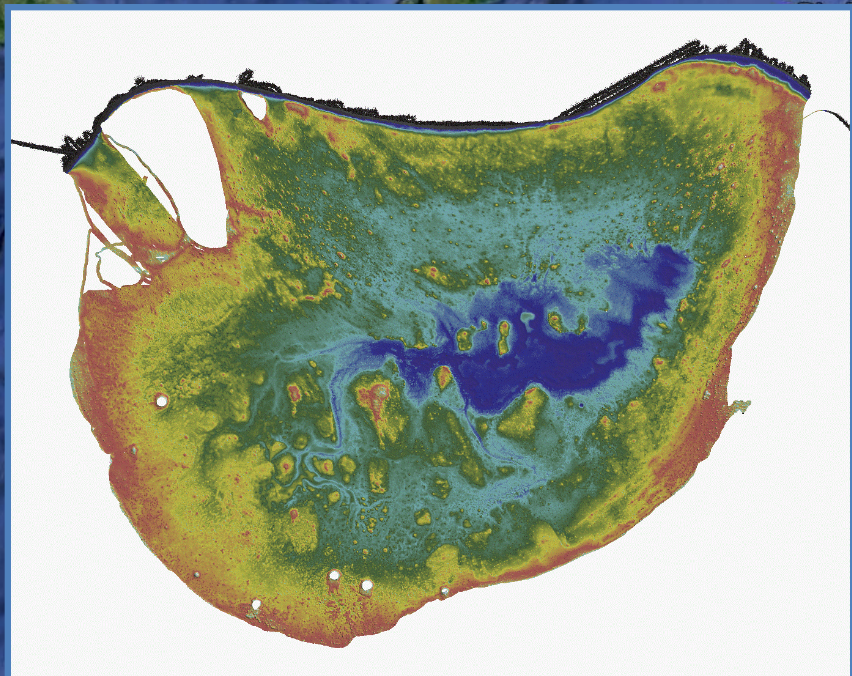
After this tutorial you will know …

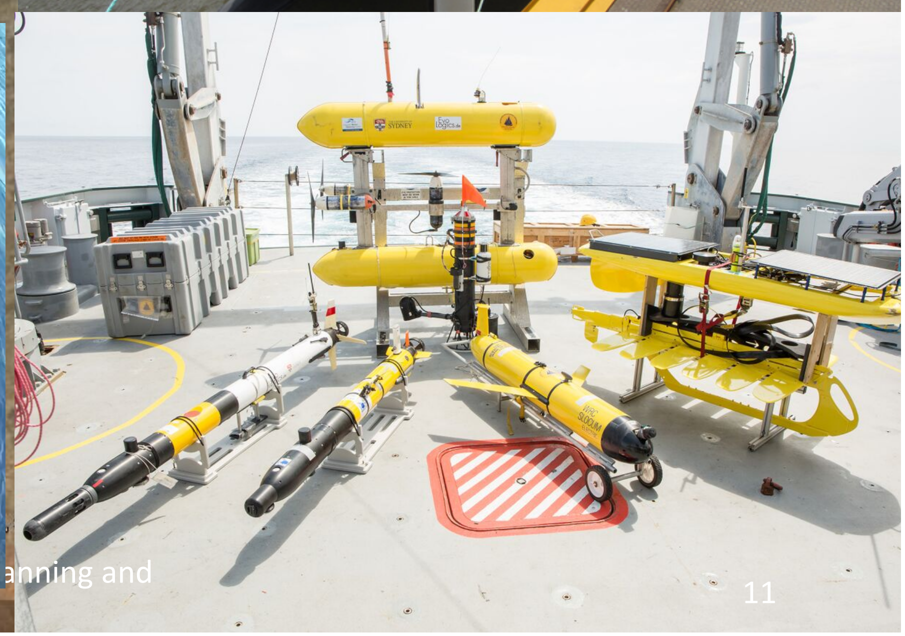
- That autonomous systems are risky,
and this risk must be managed.
- How to specify goal-directed motion planning problems as qualitative state plans (QSP).
- How to encode QSP motion planning as a
constraint optimization problem.
- When planning to maximize utility
under bounded risk makes sense.
- How to use risk allocation to solve planning and scheduling problems.
- How to perform risk allocation iteratively
and in closed form.

Outline

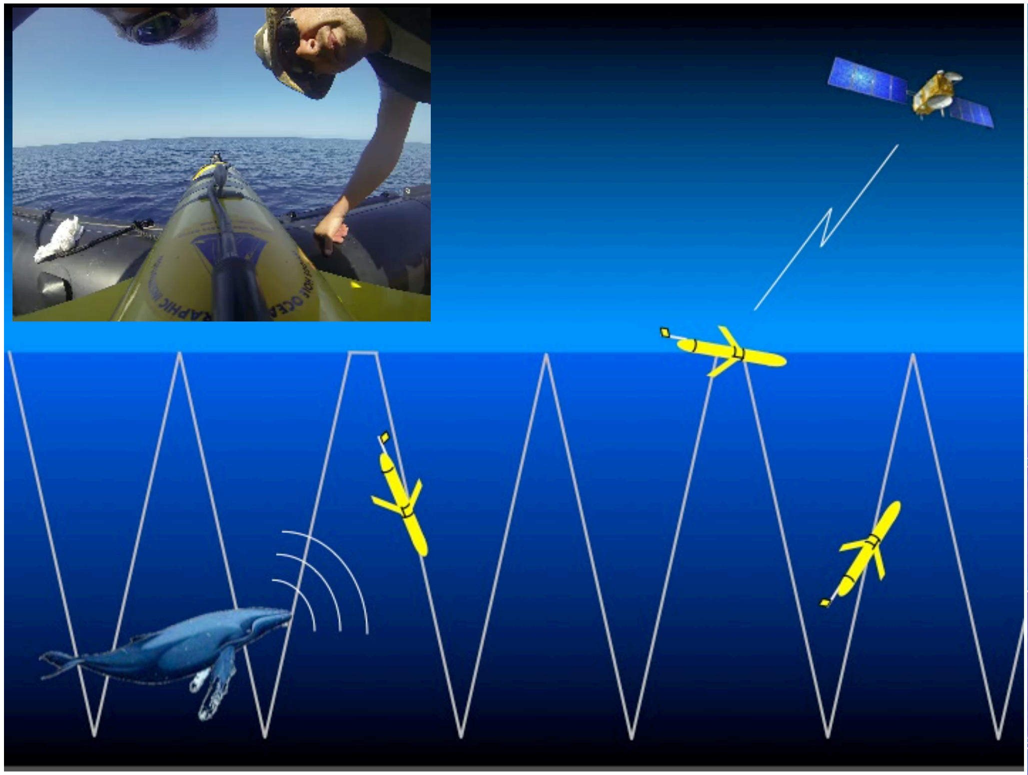
- A Risky Business
- Overview of Risk-bounded Planning
- Goal-directed Trajectory Planning
- Risk-bounded Trajectory Planning

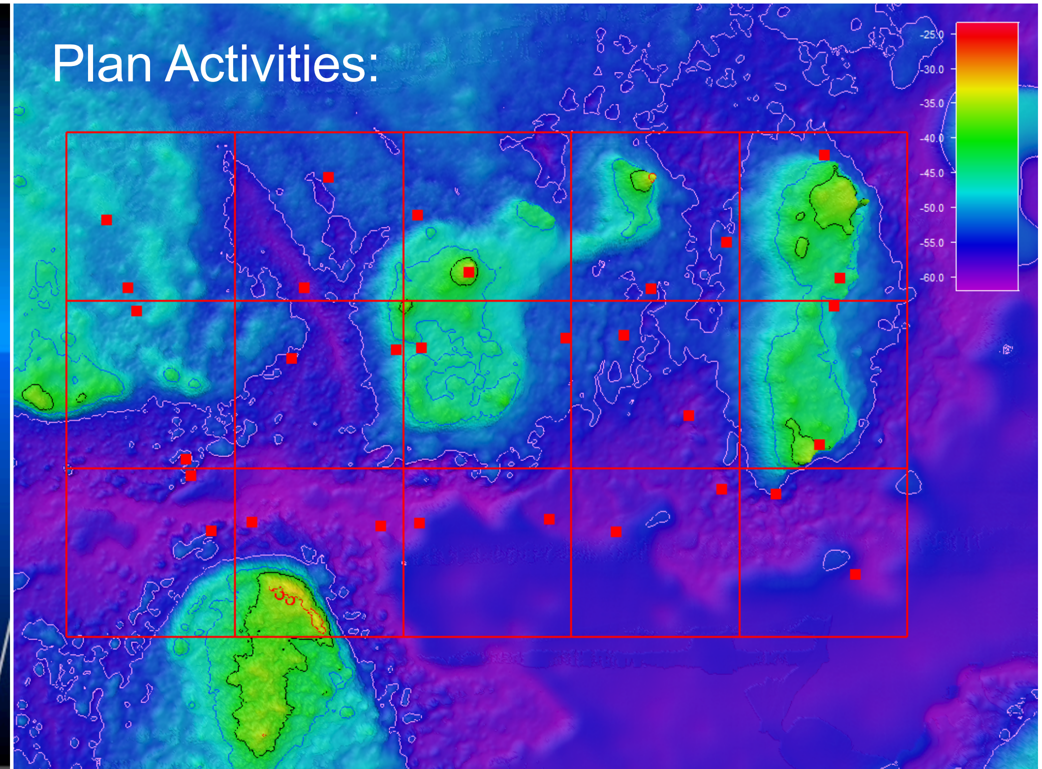
Goal-directed Planning and Execution in a Risky Environment: Falkor Cruise – March-April, 2015






Fundamental Considerations for Autonomous Gliders...
Localization
(Where am I?)
Mapping
(Where is the destination?)
Path Planning
(What is the route?)
Guidance
(Range & bearing to goal?)
Control
(Adjust actuator state?)
Scheduling
(Adequate time for task completion?)
Fault-tolerance
(Adequate system states operational?)
Coordination
(Are other robots available to assist?)
In collaboration with Dr. Rich Camilli
Deep Submergence Laboratory, WHOI



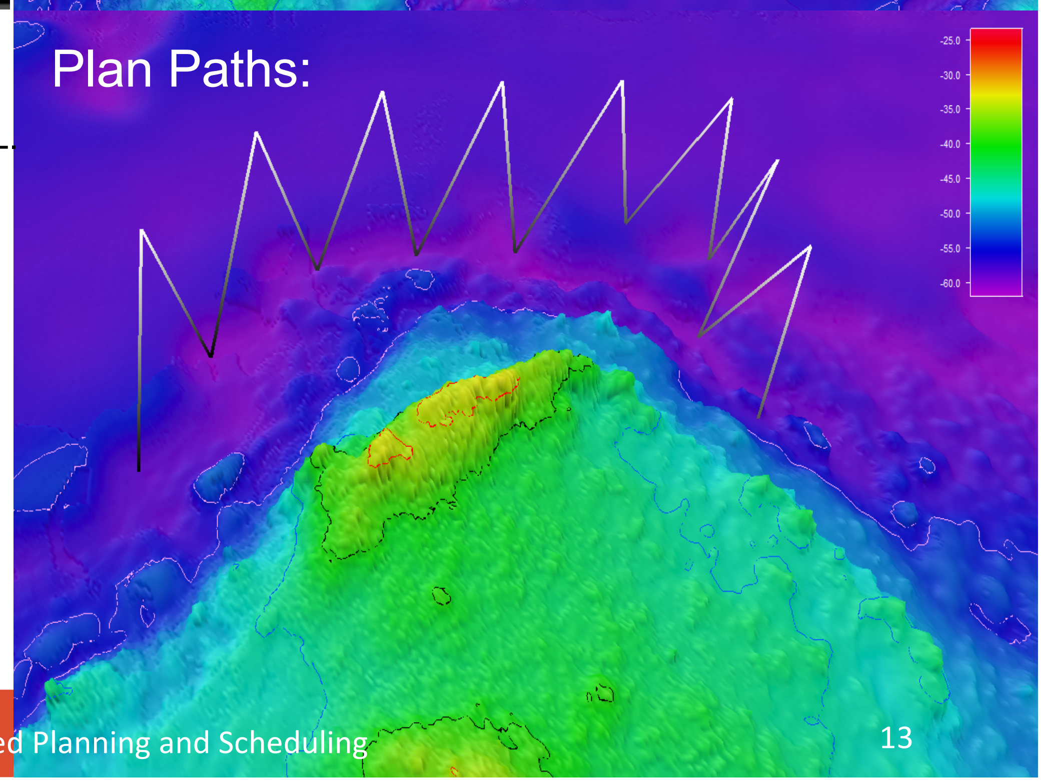


Combined Activity and Path Planning Rules:

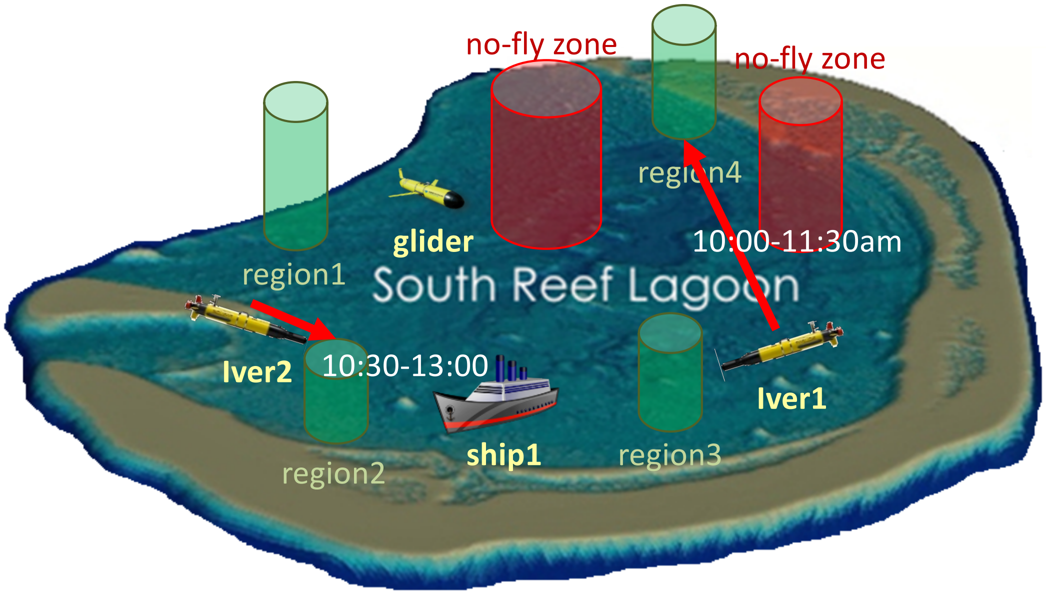
- TRANSIT THROUGH BOTH GOAL POINTS IN EACH CELL.
- AVOID FIXED AND MOVING HAZARDS.
- MINIMIZE ENERGY EXPENDITURE.
  (BY OPTIMIZE FOR: DEPTH BAND, LINEAR DISTANCE, AND TIDAL CURRENTS).
- STAY WITHIN 2KM COMMS RANGE OF MOTHER SHIP.

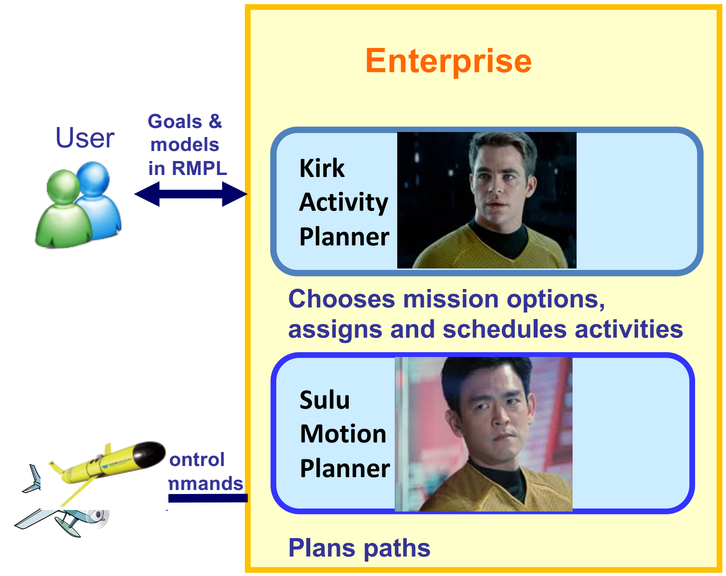
Missions Programs in RMPL:

- Parallel threads;
- Goal locations;
- Flexible time bounds;
- Decision-theoretic choice.
- Safety margins, ⇒ more recently risk-bounds.



Involves Coupled Activity and Motion Planning





Program A ::=
prim_action(args) |
remain_in(R) | start_in(R) | end_in(R) |
[lb, ub] A |
Sequence {A1; A2; … } |
Parallel {A1; A2; … } |
Choose { [with reward R1] A1;
[with reward] R2 A2; . . .} |

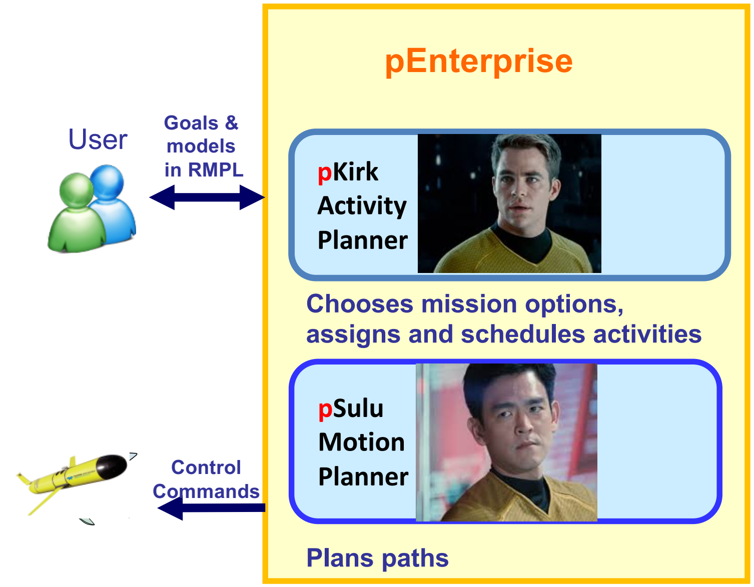
Model:

- Durative PDDL
- $X_{t+1} = Ax_t + Bu_t$
- Topographical Map
- Target Regions
- Currents

Involves Risk-bounded Activity and Motion Planning

pKirk: Uses pSTN schedulers (Picard/ Rubato).

pSulu: Uses iterative or non-convex risk allocation
to handle obstacles.




```
Program A ::=
    prim_action(args)    |
    remain_in(R) | start_in(R) | end_in(R) |
    [lb, ub] A   |
    ~N(mu, sigma) A  |
    Sequence {A1; A2; ... } |
    Parallel {A1; A2; ... }     |
    Choose { [with reward R1] A1;
                  [with reward] R2 A2; . . .}  |
chance_constraint: Delta A
```

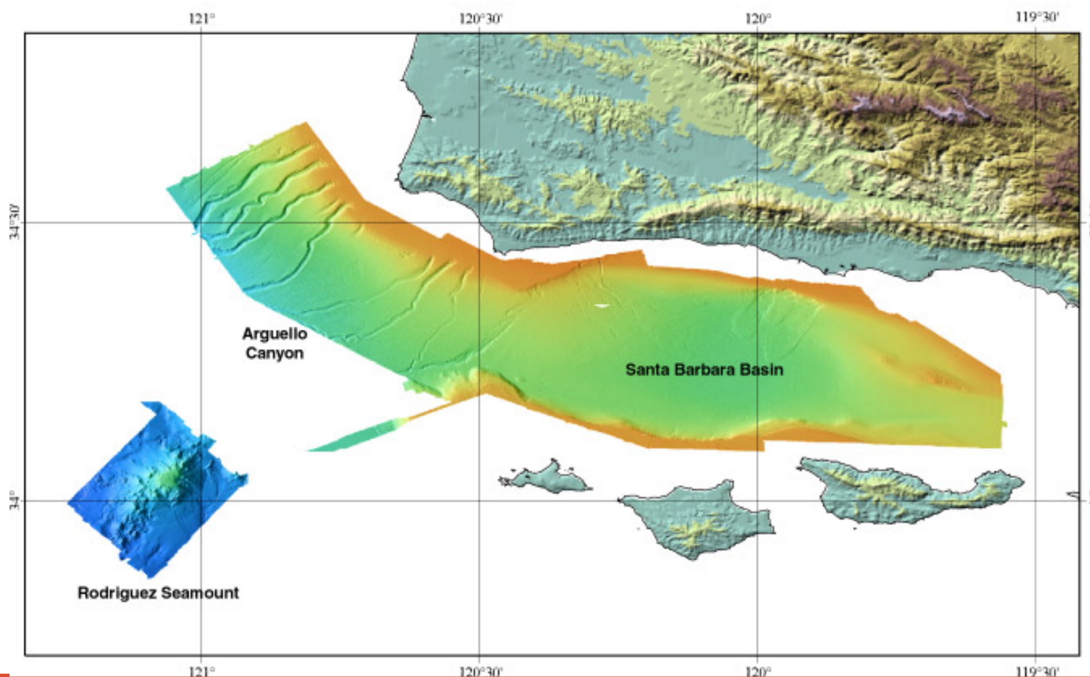
Model:

- Durative PDDL
- $X_{t+1} = Ax_t + Bu_t + w_t$

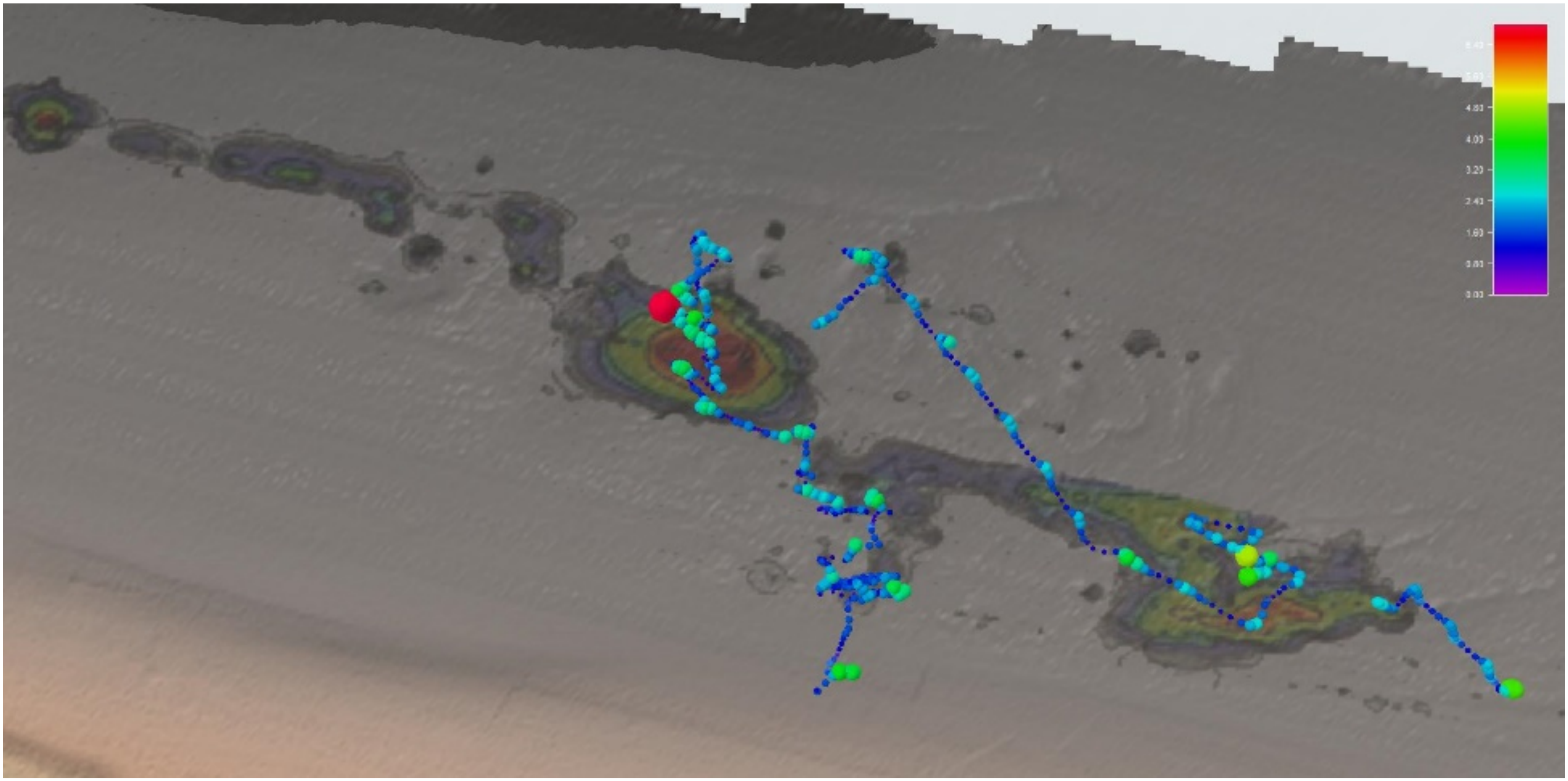
- Uncertain Topo Map
- Target Regions
- Uncertain Currents

September, 2017 :
Slocum Glider(s) at Santa Barbara

- Mission goal:
  - Use miniaturized mass spectrometer to find and characterize oil seeps off the coast of Santa Barbara.
- Research goal:
  - Combined, risk-aware activity and motion planning.



Observed Benzene Concentration

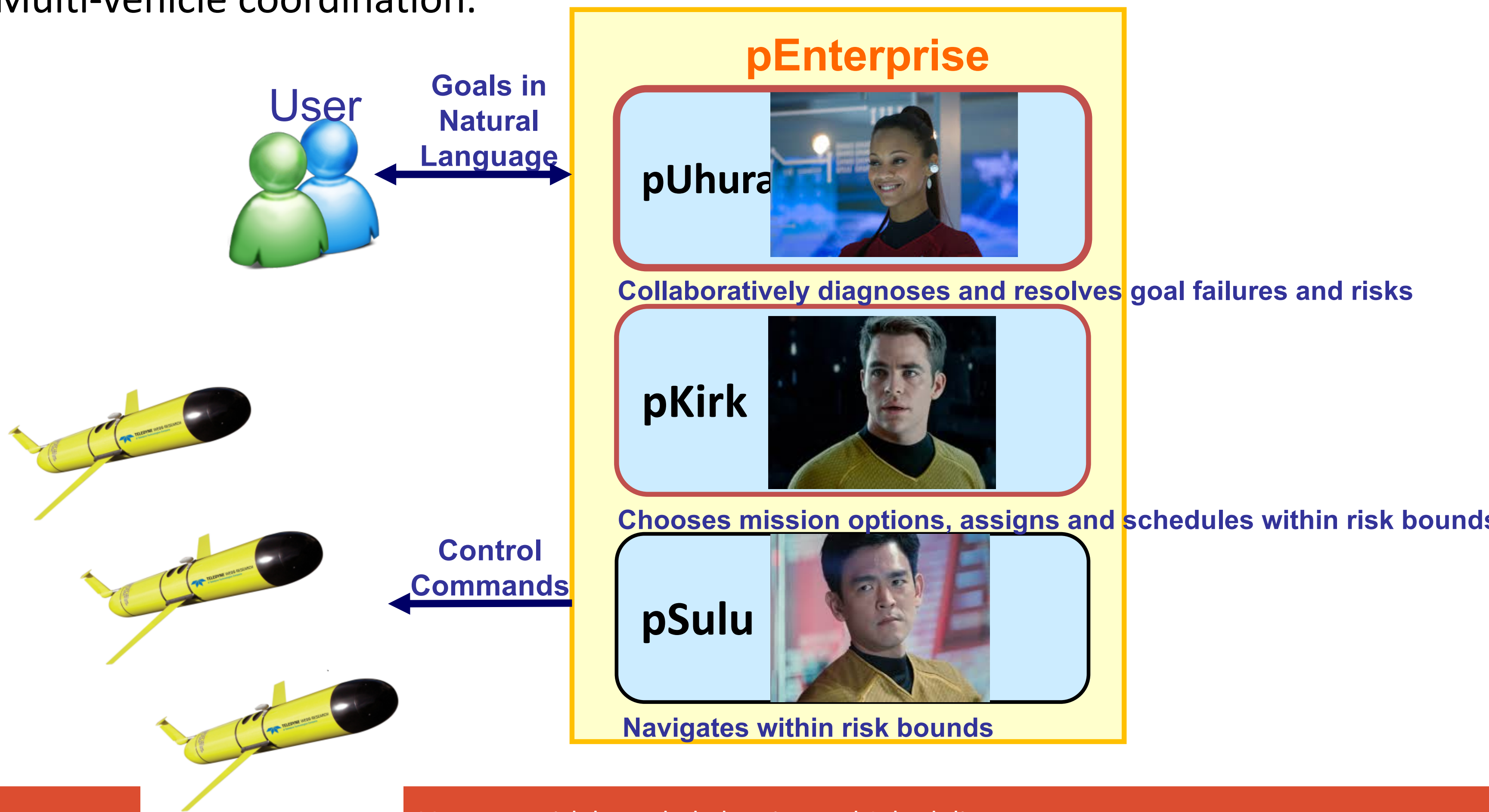


October, 2017 :
Slocum Gliders off Cape Cod

Mission goal: Sustained operation of network

Research goals:

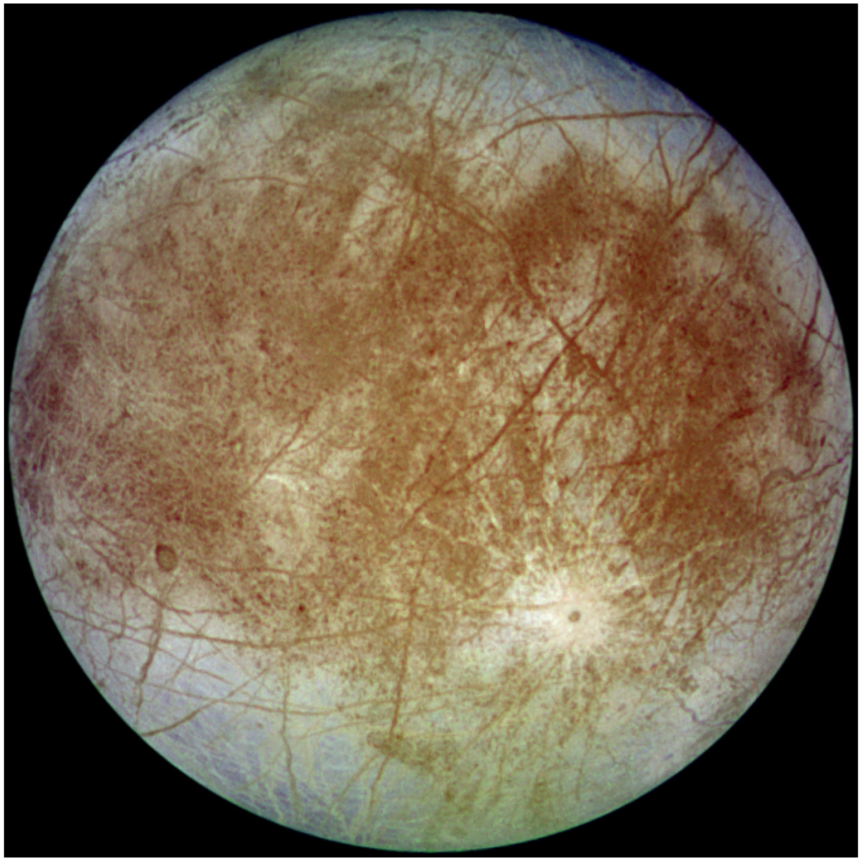
- pUhura: Risk-bounded Relaxation.
- pKirk: Multi-vehicle coordination.



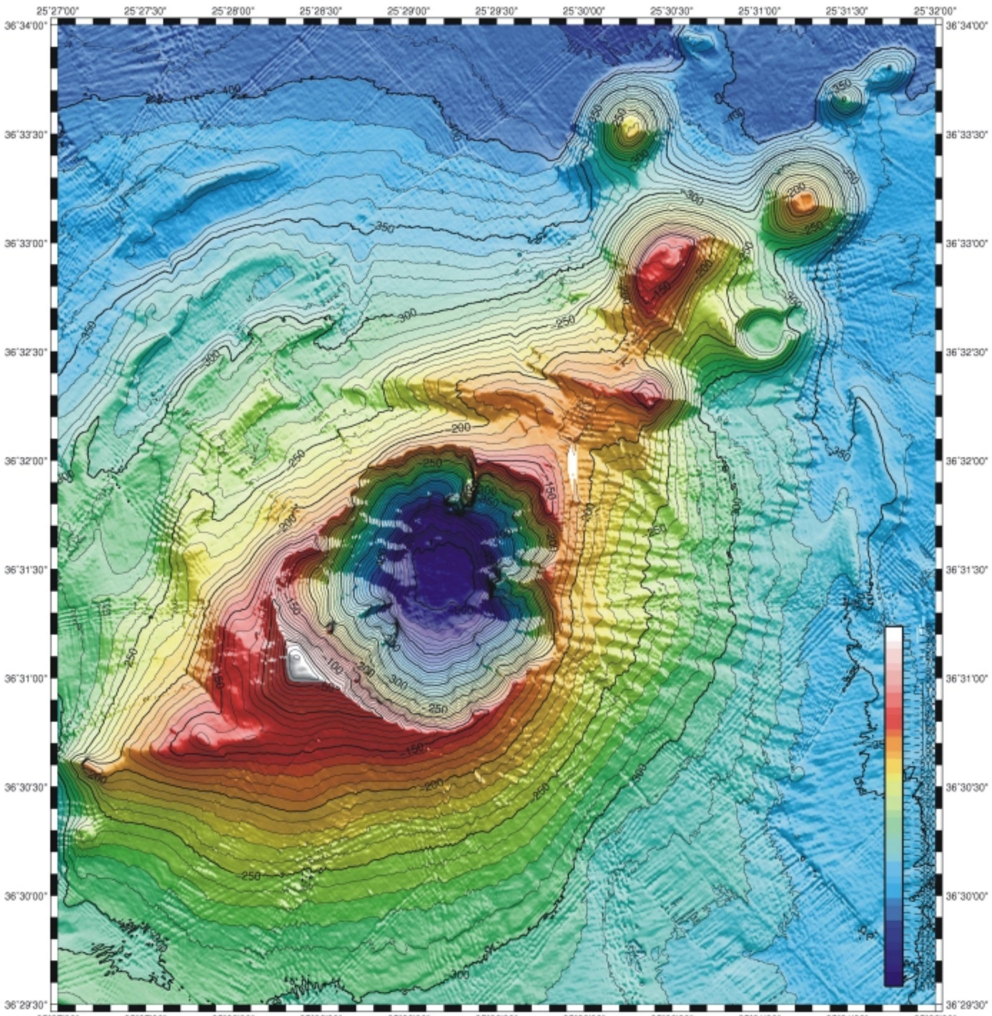
Risk-bounded, autonomous, deep sea exploration

on Earth

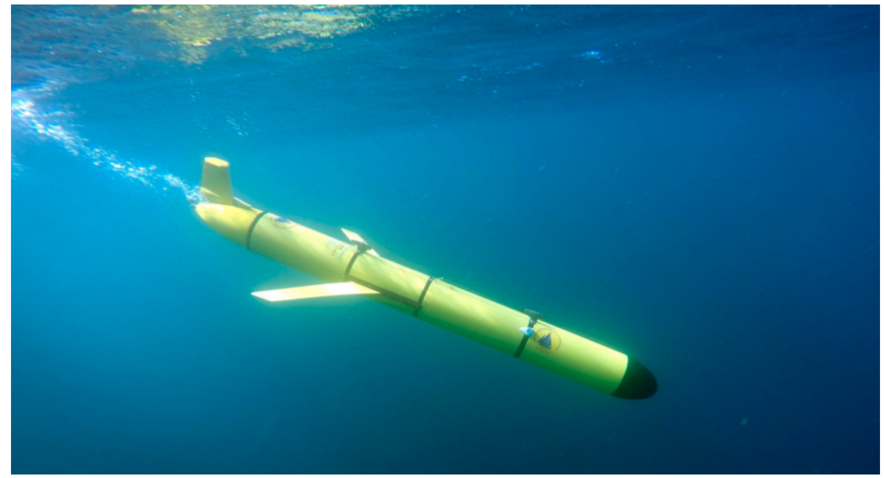
and other planets and moons

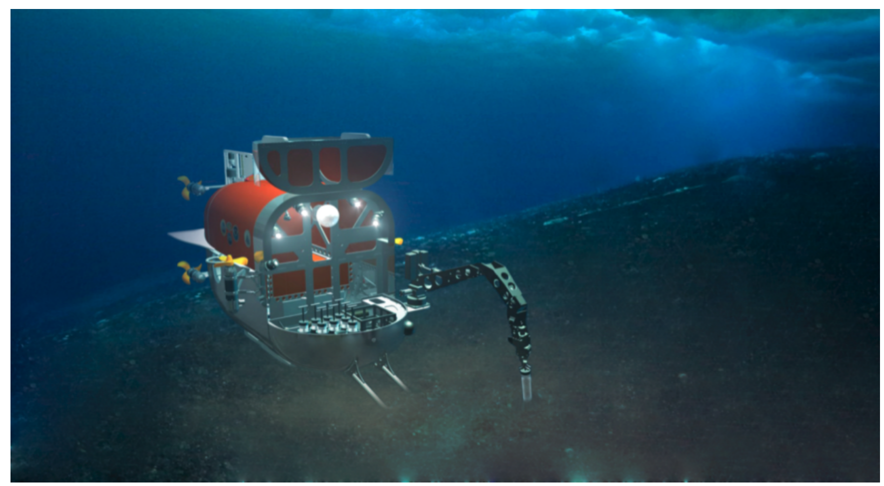


2019: Europa Analog Mission Demonstration Of Risk-bounded, Autonomous Exploration



Mission: Look for evidence of “extreme” life at Kolumbo Deep-Sea Volcano near Santorini, Greece

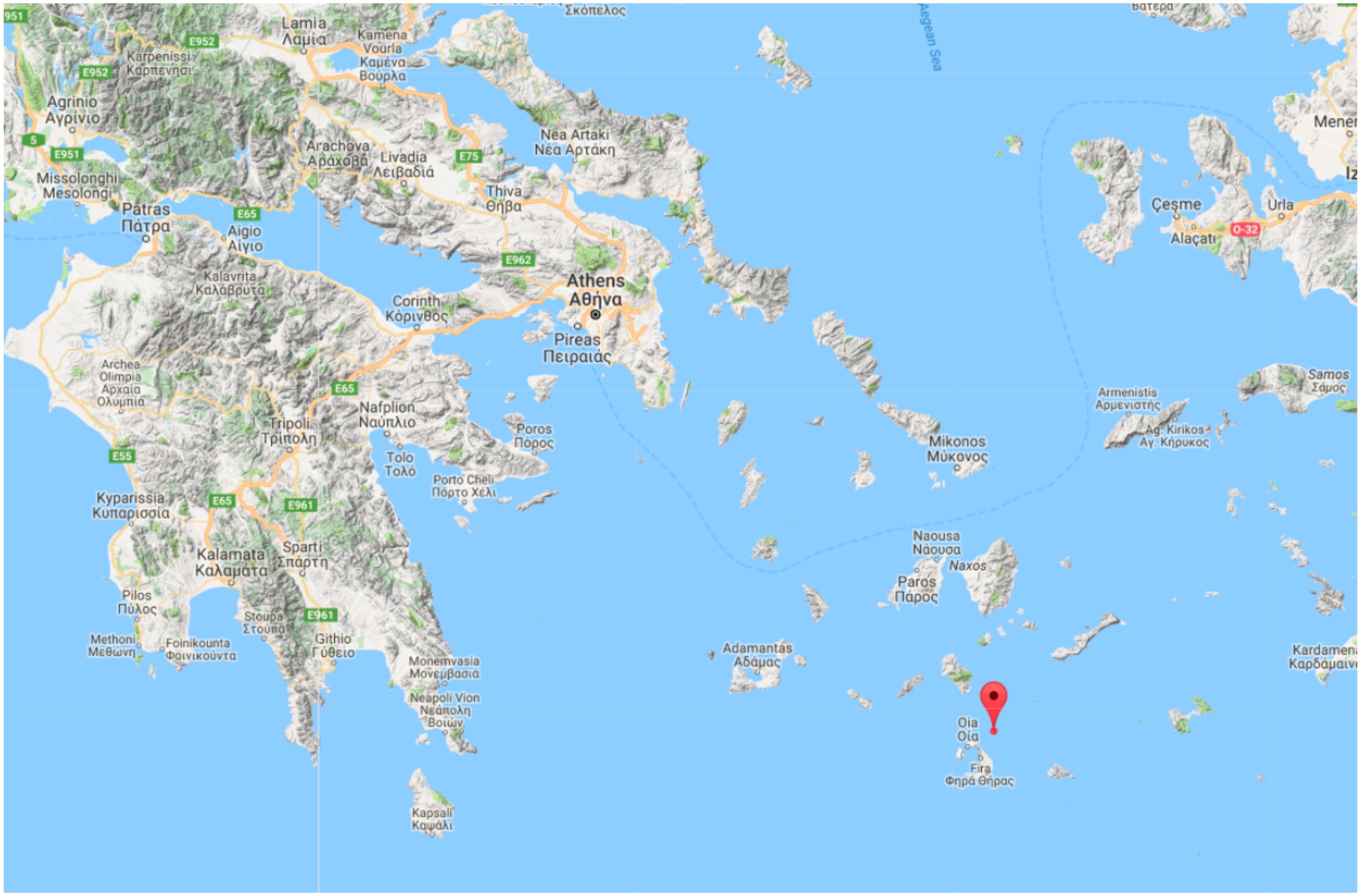




funded by NASA PSTAR Program

Team: WHOI, MIT, ACFR, U. Michigan

Kolumbo Location



8 km northeast of Santorini Island, Greece

Kolumbo Volcano Mission

- NASA Analog Mission

  Autonomous underwater exploration with heterogeneous coordinated vehicles (e.g. Europa moon)

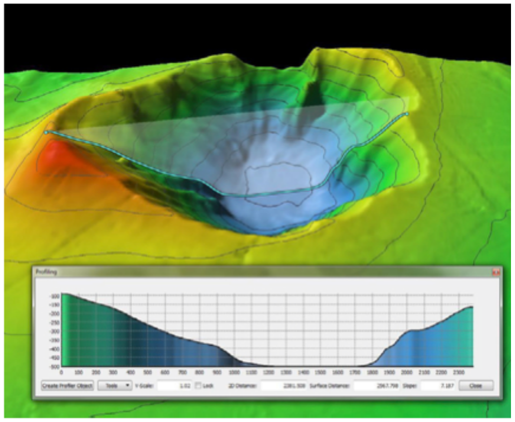
- Science Goals

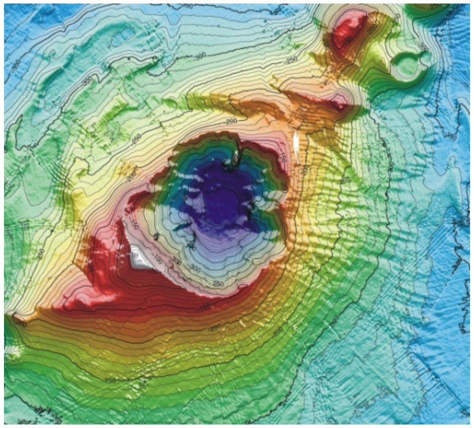
  Explore the **Kolumbo caldera**

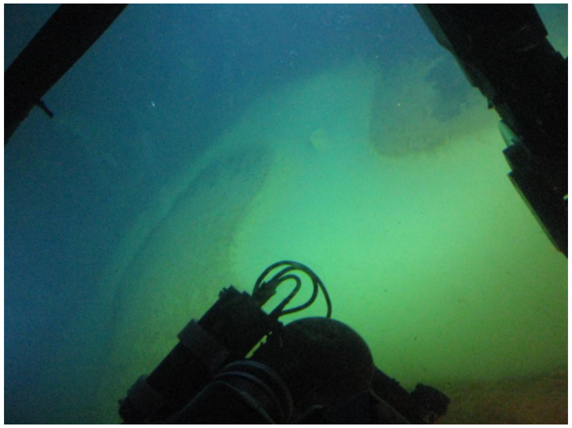
  – Relatively unexplored

  – Biological and geological interest due to volcanic activity

    - previously undiscovered CO_2 pools in the region and known hydrothermal vents

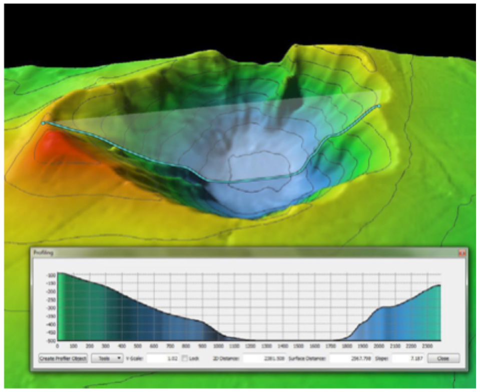


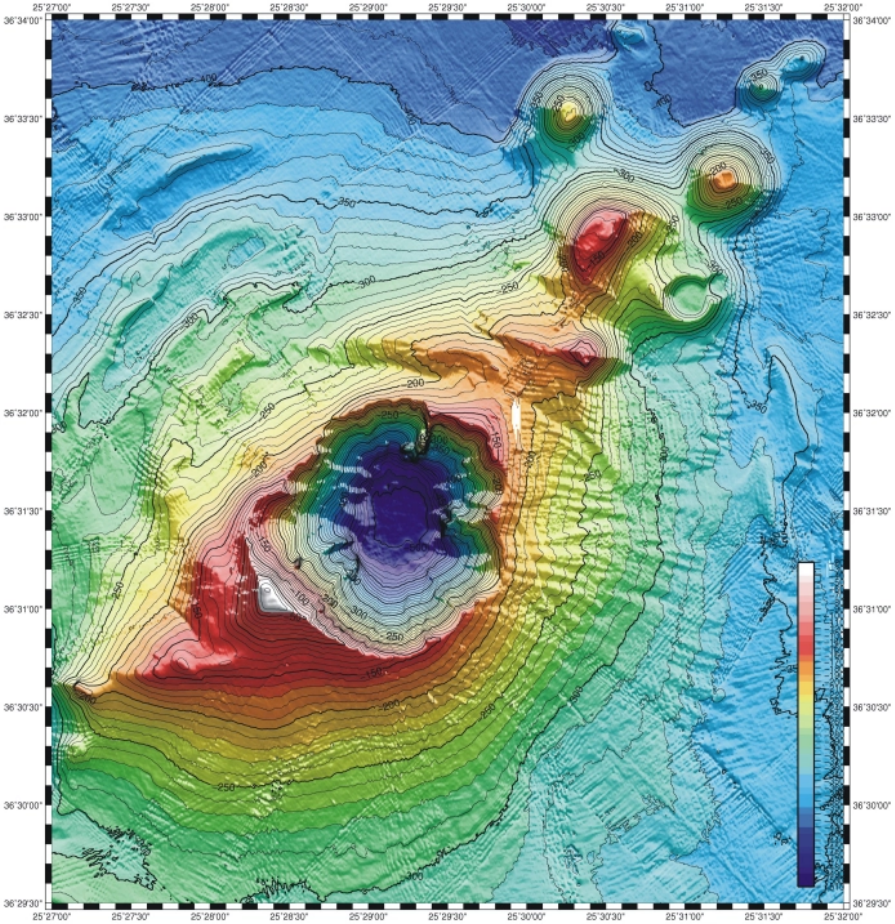




Kolumbo Caldera

- ~7km diameter
- Shallowest (rim) at 18m depth
- Deepest at ~500m
- Risky environment
  - Currents (vertical and horizontal)
  - Sharp slope
  - Overhang obstacles
- Locations of science sites unknown



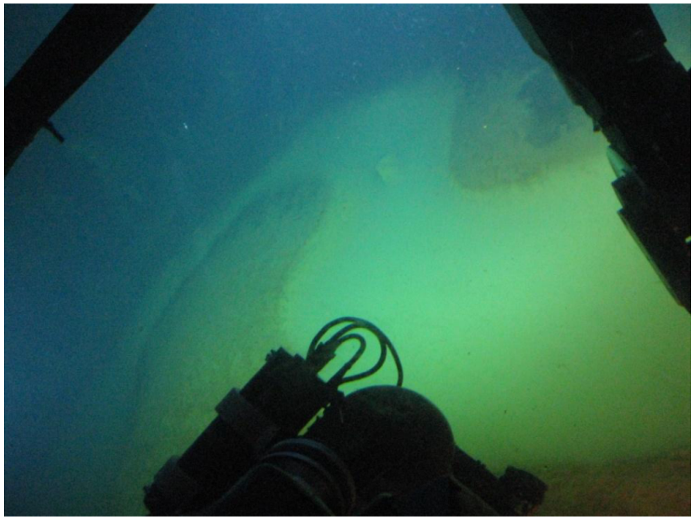


Biological Science Goals

Search for life forms that flourish in extreme hypercapnia, within CO_2 accumulating subsea pools (*Kallisti Limnes - Camilli*).

Offers Insights into:

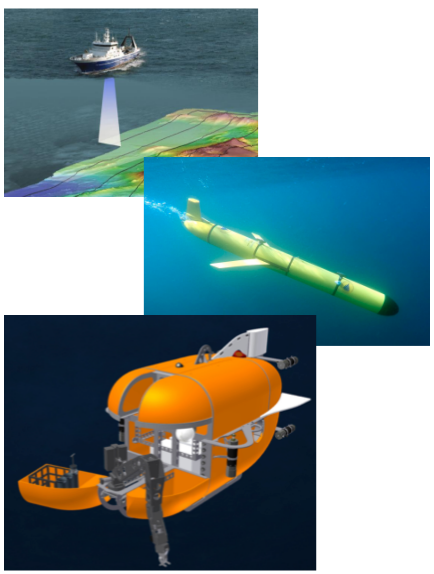
- biological context of Earth's precambrian evolution towards an oxidizing atmosphere
- possible life on other planetary bodies where sunlight is not available.



Camilli et al, *Nature Scientific Reports* 2015

Autonomous Planetary Exploration Goals

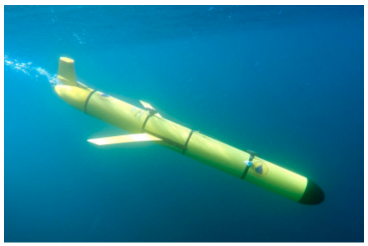
- Demonstrate new methods for exploration, using coordinated heterogeneous vehicle platforms that:
  1. rely upon automated mission re-planning at all levels
  2. incorporate environmental and vehicle data to maximize information gain, and
  3. respond to risk, uncertainty and evolving science goals

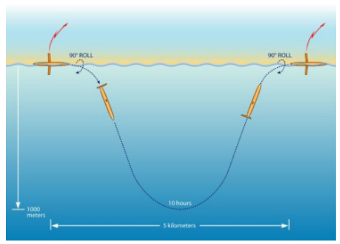


1. **Surface vessel (ship) →** "remote sensing orbiter"
   - Will map the volcanic crater using acoustic remote sensing
2. **AUV glider →** "long-range reconnaissance drone"
   - Uses underwater mass spectrometer to localize chemical anomalies
3. **Nereid Under Ice** (Hybrid ROV) **→** "lander"
   - Employs sonar, stereo cameras, chemical sensors, manipulator arm to characterize and retrieve sample.

Operational Challenges of Heterogeneous Platforms

Glider



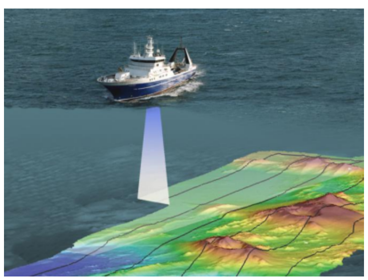


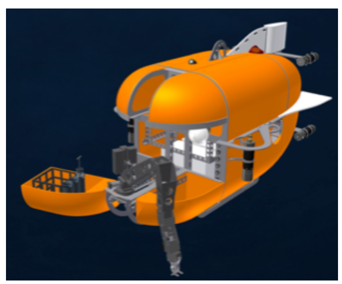
Mode switch of hybrid glider mode changes dynamics:

- Buoyancy engine
- Thruster

Can only communicate at the surface

ROV + Ship





100-200m horizontal ROV excursion from ship

- Joint navigation of both ship and ROV

Manipulator operates on steep terrain:

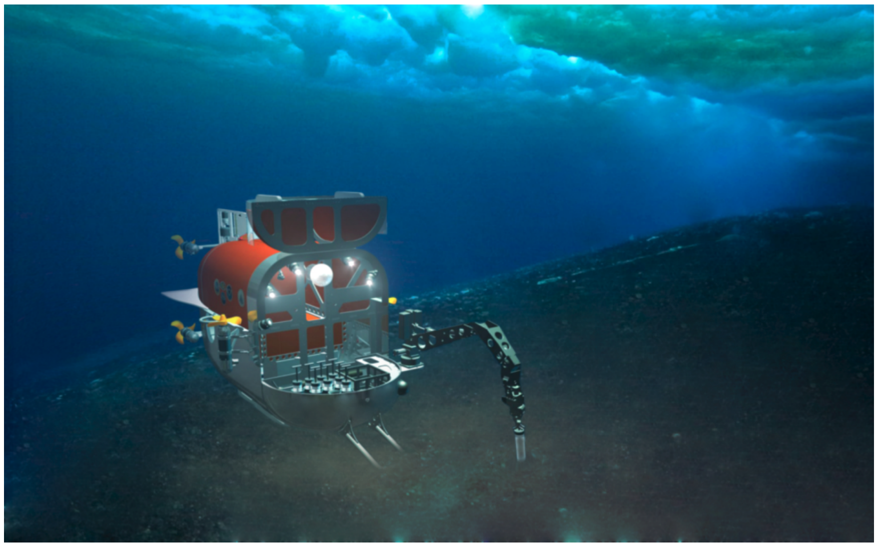
- Coordinated motion planning of vehicle and arm during sampling

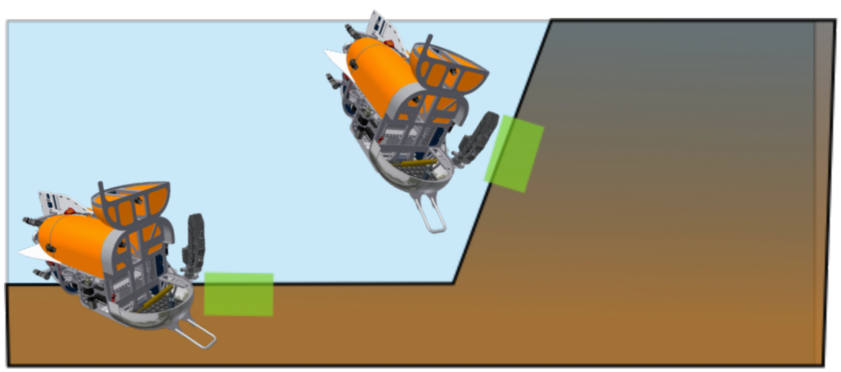
Resource constraints:

- 6-8 hour battery underwater
- Manipulator usage affects battery

Collision avoidance and coordination between all of them

Challenges of Motion Planning and Manipulation



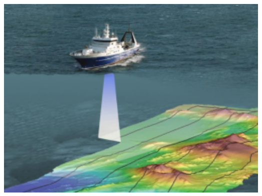


Sensing: 3D scene reconstruction with stereo cameras

Sampling while landed:
- ROV body attached to the seafloor
- Arm moves independent of body

Sampling while hovering:
- No suitable attachment point
- ROV’s arm and body motion coupled
- Disturbances due to currents

Mission Overview

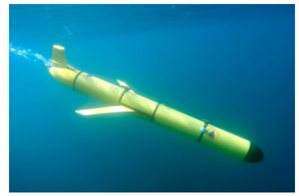


Stage 1

High-res multibeam scan (~ 1 day)
~ 1 month earlier

Scientists compose **prior** for regions of interest:

1. Previous low-res multibeam
2. Historical known hydrovents
3. Volcanologist interpretation of high-res multibeam



Stage 2: AUV Glider

~ 1 week

Offline
Compute plan that maximizes prior

Online
Glider executes plan but performs adaptive sampling and **re-plans** as required to maximize information gain

- Mission safety
- Temporal & position constraints

Stage 3: ROV + Glider

~ 1 week

Experts decide most promising goals for ROV **inspection and sampling**.

Glider **continues adaptive exploration** and informs ROV.

Multi-vehicle planning (ship, ROV & glider) to maximize scientific return while respecting constraints.

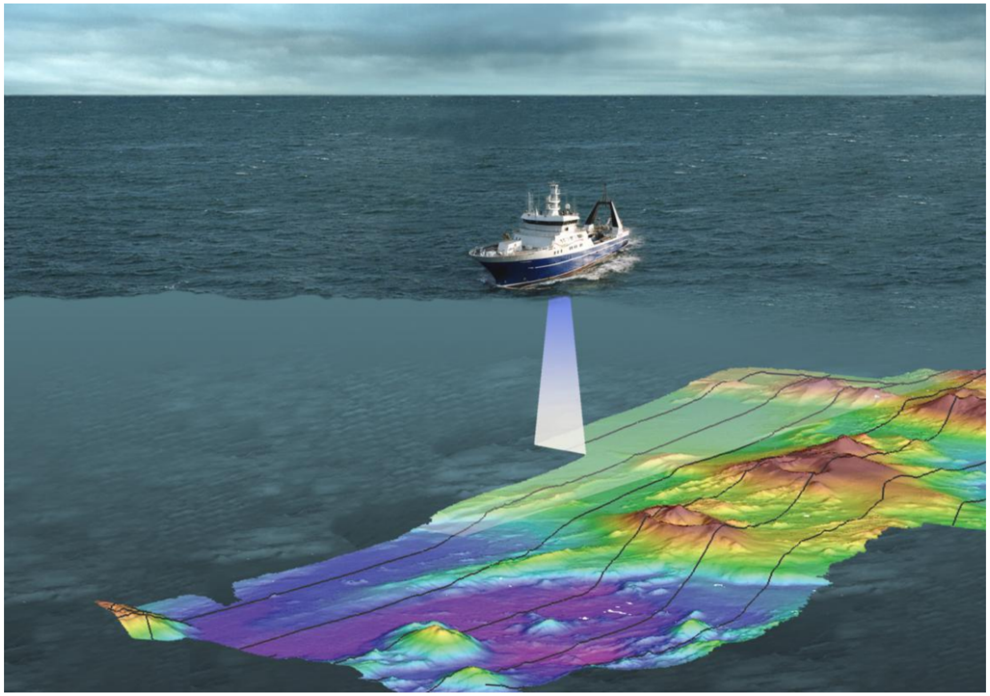
- Distance between ship & ROV, Glider
- ROV battery constraints

Stage 1: Ship Survey

- Maps volcano seafloor with multi-beam
- Analyzes multi-beam data for regions of interest.

Output:

- High-res multibeam data
- Interpretation made by scientist (**prior** for regions of interest)

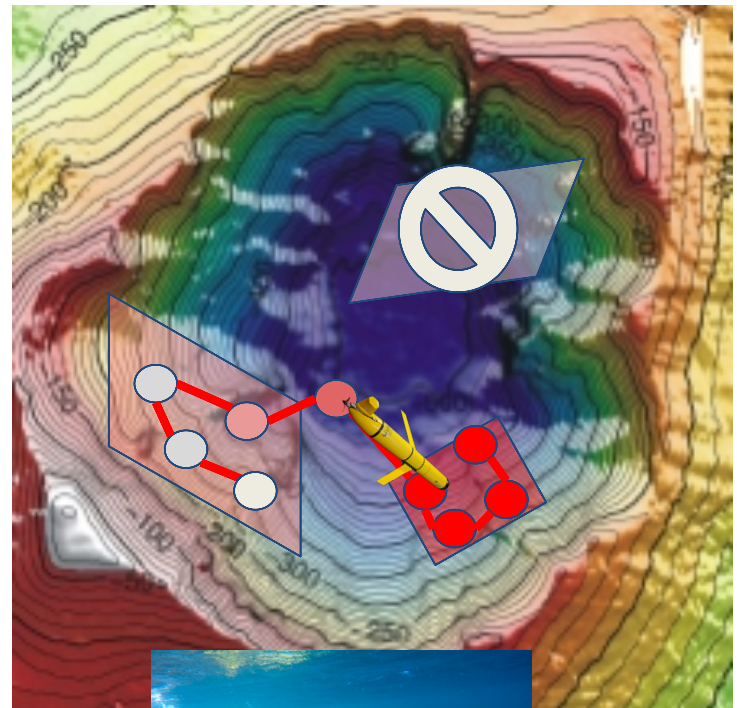


When: About one month before Stage 2

Picture from Schmidt Ocean Institute. https://schmidtocean.org/cruise-log-post/mapping-earths-ocean-seafloor/

Stage 2: Adaptive Search w/ Glider

Glider adaptively plans sequences of visitations that maximize expected information gain, while bounding risk.



Requires:
- Risk-bounded adaptive sampling planner
- Effective planning for different velocity modes and configurations of the hybrid AUV glider
- Coupling to terrain-aided navigation with sonar

Output:
- Chemical analysis of water column samples
- **Posterior** for regions of interest, based on chemical indicators
- Promising locations for further investigation and sampling by ROV

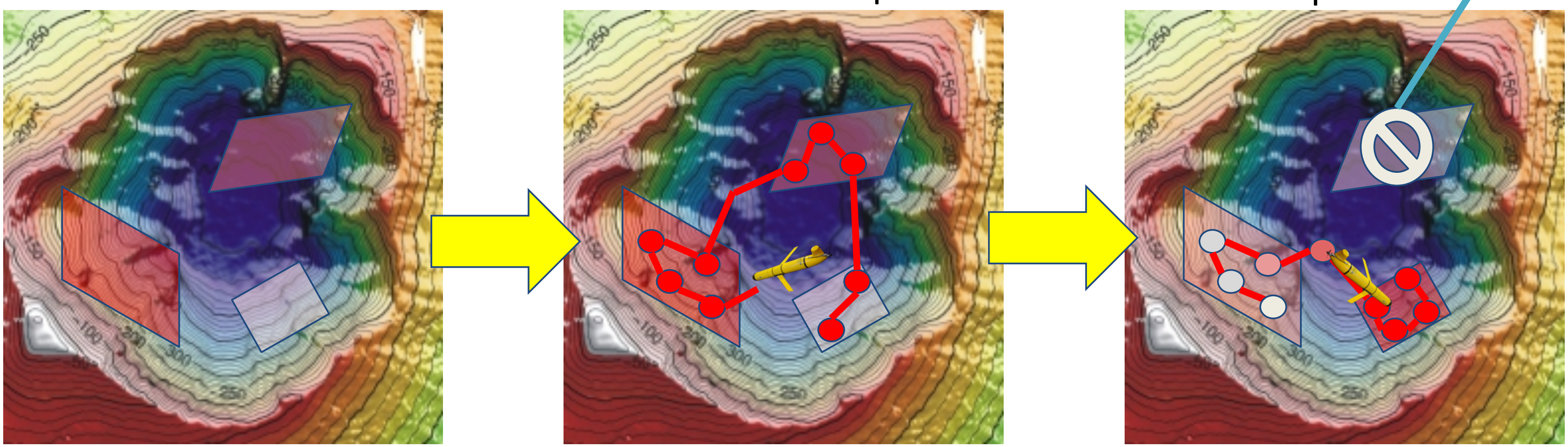
Stage 2: Adaptive Search w/ Glider

"**Prior**" for interesting regions
as decided by the scientists
based on surface vessel data

Initial offline plan

Online adaptation

Belief updated .
no time left to explore
less promising region



Utilizing **risk-bounded** information maximization algorithms.

Stage 3: ROV Sampling + Glider Adaptive Search

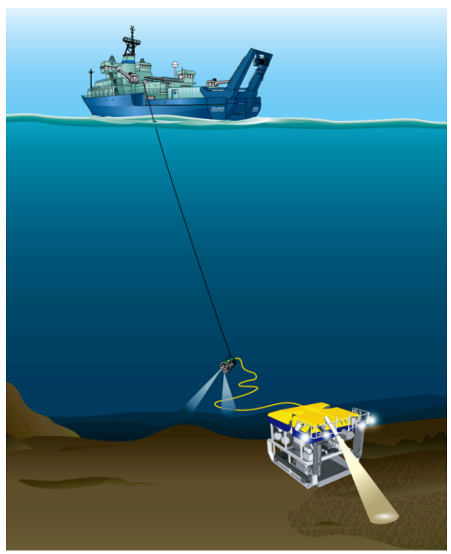
ROV inspects regions and collects samples, while Glider simultaneously explores.

Requires:

- Joint navigation of ship and ROV
- Simultaneous navigation of ship/ROV and glider, under limited communication.
- Joint motion planning of ROV body and arms

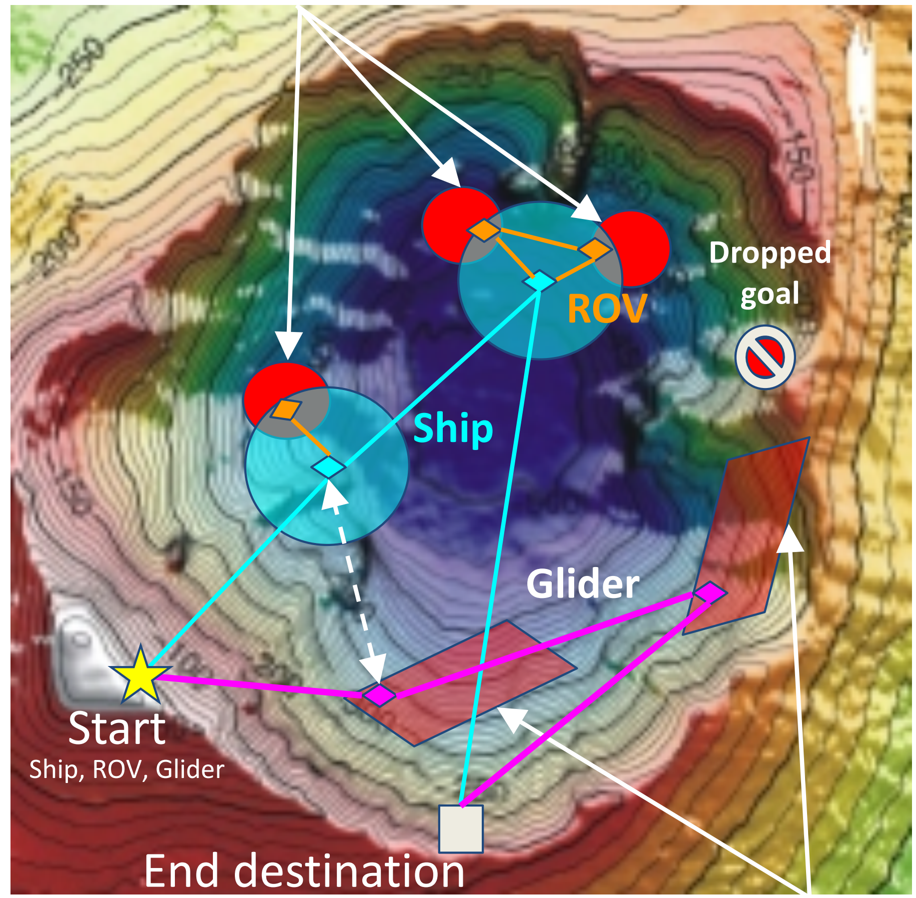
Output:

- Collected samples
- Science data



Graphic of research vessel, Medea and Jason, from WHOI, http://www.whoi.edu/page.do?pid=80696&i=17162

Desired ROV sampling locations

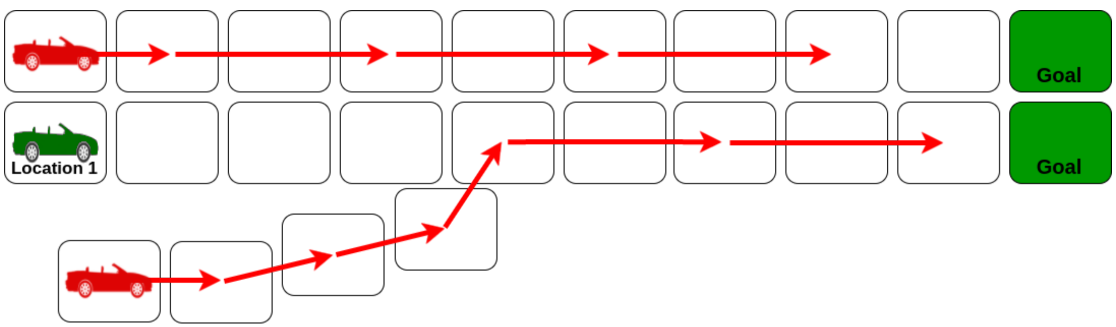


Areas of most interest for glider

Outline

- A Risky Business
- Overview of Risk-bounded Planning
  - Risk-bounded Probabilistic Planning
  - Risk-bounded Trajectory Planning
  - Risk-bounded Scheduling
- Goal-directed Trajectory Planning
- Risk-bounded Trajectory Planning

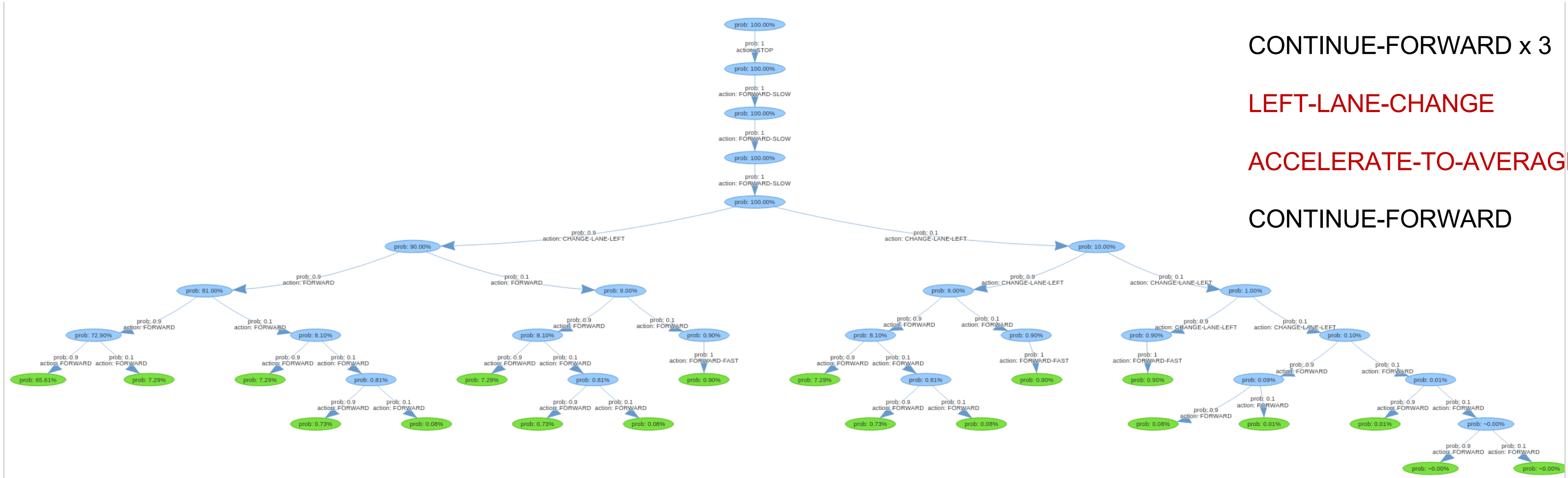
Pre-planning Maneuvers on the Highway



Our driver in the right lane,
a vehicle entering on ramp,
and a passing vehicle.

Risk of near collision bounded to 0.001.

Resulting Conditional Plan



Most Likely Maneuver Sequence:

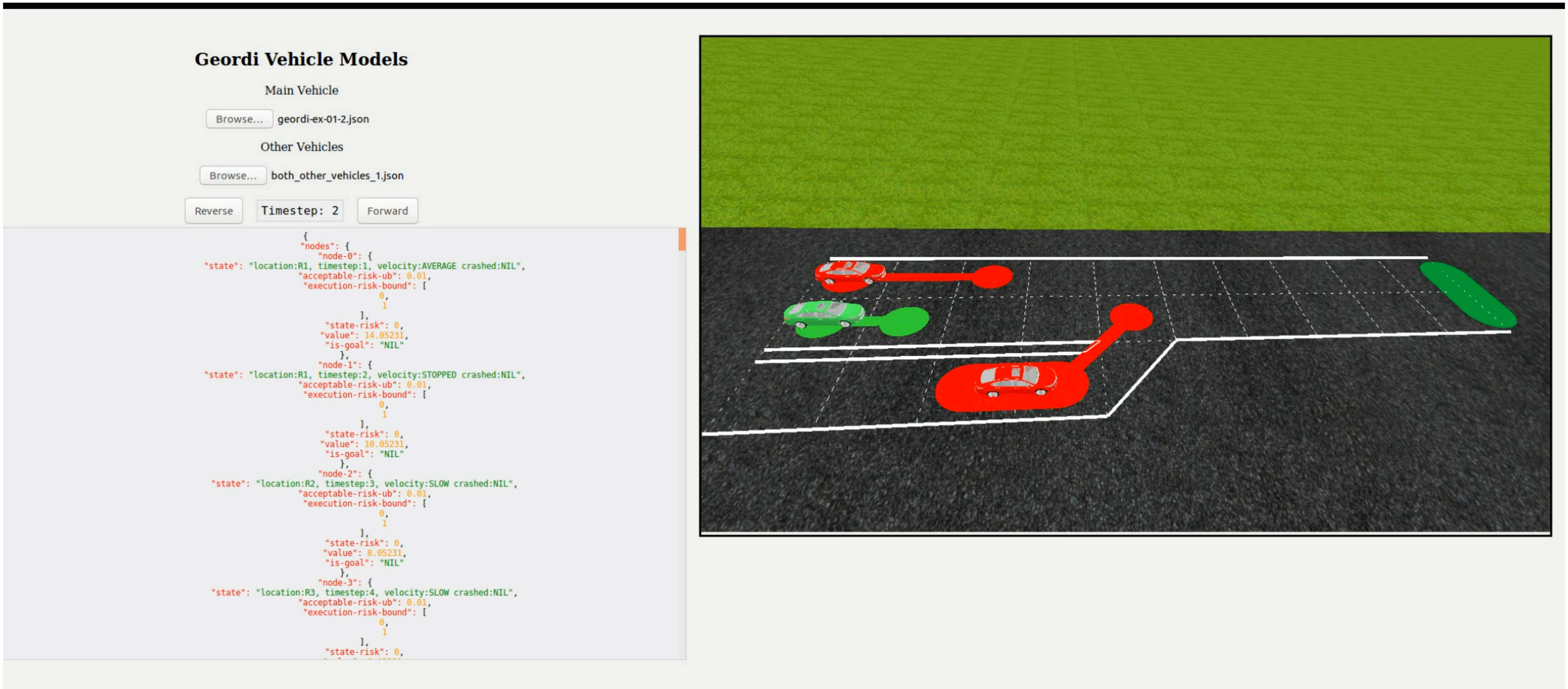
DECELERATE-TO-SLOW

CONTINUE-FORWARD x 3

LEFT-LANE-CHANGE

ACCELERATE-TO-AVERAGE

CONTINUE-FORWARD


Geordi Vehicle Models
Main Vehicle
Browse...
geordi-ex-01-2.json
Other Vehicles
Browse...
both_other_vehicles_1.json
Reverse
Timestep: 2
Forward
{
"nodes": {
"node-0": {
"state": "location:R1, timestep:1, velocity:AVERAGE crashed:NIL",
"acceptable-risk-ub": 0.01,
"execution-risk-bound": [
0,
1
],
"state-risk": 0,
"value": 14.05231,
"is-goal": "NIL"
},
"node-1": {
"state": "location:R1, timestep:2, velocity:STOPPED crashed:NIL",
"acceptable-risk-ub": 0.01,
"execution-risk-bound": [
0,
1
],
"state-risk": 0,
"value": 10.05231,
"is-goal": "NIL"
},
"node-2": {
"state": "location:R2, timestep:3, velocity:SLOW crashed:NIL",
"acceptable-risk-ub": 0.01,
"execution-risk-bound": [
0,
1
],
"state-risk": 0,
"value": 8.05231,
"is-goal": "NIL"
},
"node-3": {
"state": "location:R3, timestep:4, velocity:SLOW crashed:NIL",
"acceptable-risk-ub": 0.01,
"execution-risk-bound": [
0,
1
],
"state-risk": 0,

Risk-bounded Probabilistic Planning (RAO*,iDual)

Ideas:

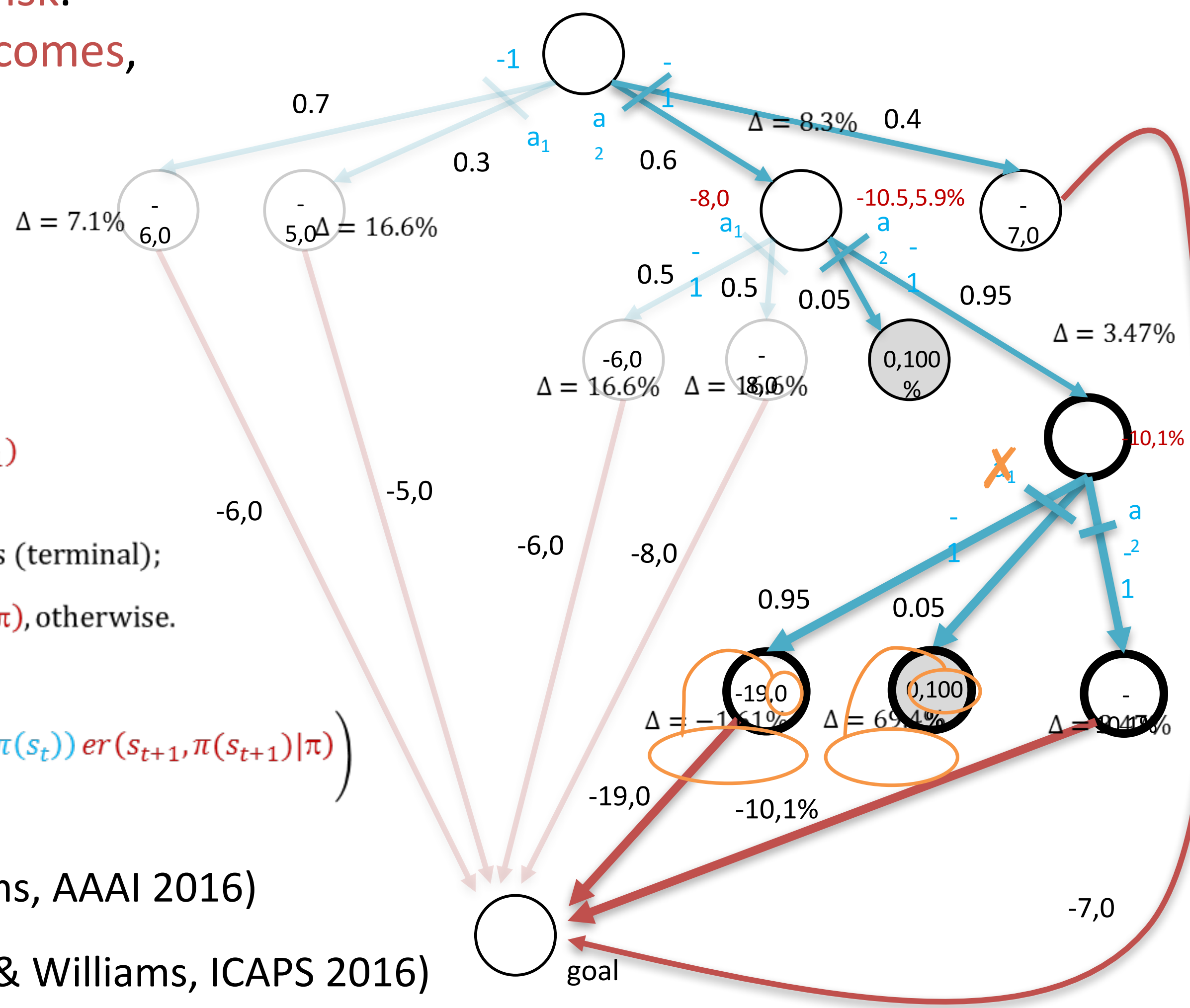
- Prunes plans with excessive risk.
- Prunes plans for unlikely outcomes, once risk-bounds met.

Risk-bounded AO* Backup:

$$Q^*(s_t) = \min_{a \in A(s_t)} c(s_t, a) + \sum_{s_{t+1}} \Pr(s_{t+1} | s_t, a)\, Q^*(s_{t+1})$$

$$er(s_t|\pi) = \begin{cases} 1, & \text{if } s_t \text{ violates constraints (terminal);} \\ \sum_{s_{t+1}} \Pr(s_{t+1}|s_t, \pi(s_t))\, er(s_{t+1}, \pi(s_{t+1})|\pi), & \text{otherwise.} \end{cases}$$

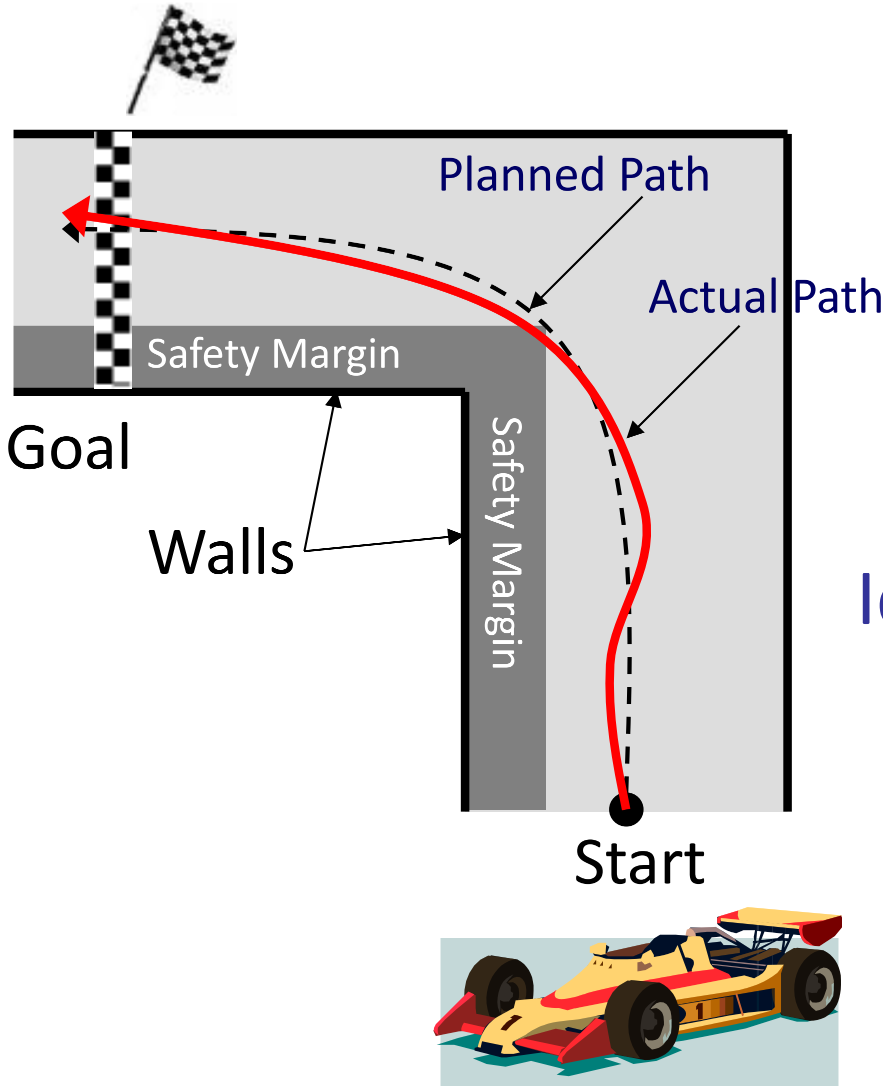
$$\Delta'_{t+1} = \frac{1}{\Pr(s'_{t+1}|s_t, \pi(s_t))}\left(\Delta_t - \sum_{s_{t+1} \neq s'_{t+1}} \Pr(s_{t+1}|s_t, \pi(s_t))\, er(s_{t+1}, \pi(s_{t+1})|\pi)\right)$$



RAO* (Santana, Thiebaux & Williams, AAAI 2016)

iDual (Trevizan, Thiebaux, Santana & Williams, ICAPS 2016)

Risk-bounded Path Planning



Problem

Find the fastest path to the goal, while limiting the probability of crash throughout the race *to 0.1%*

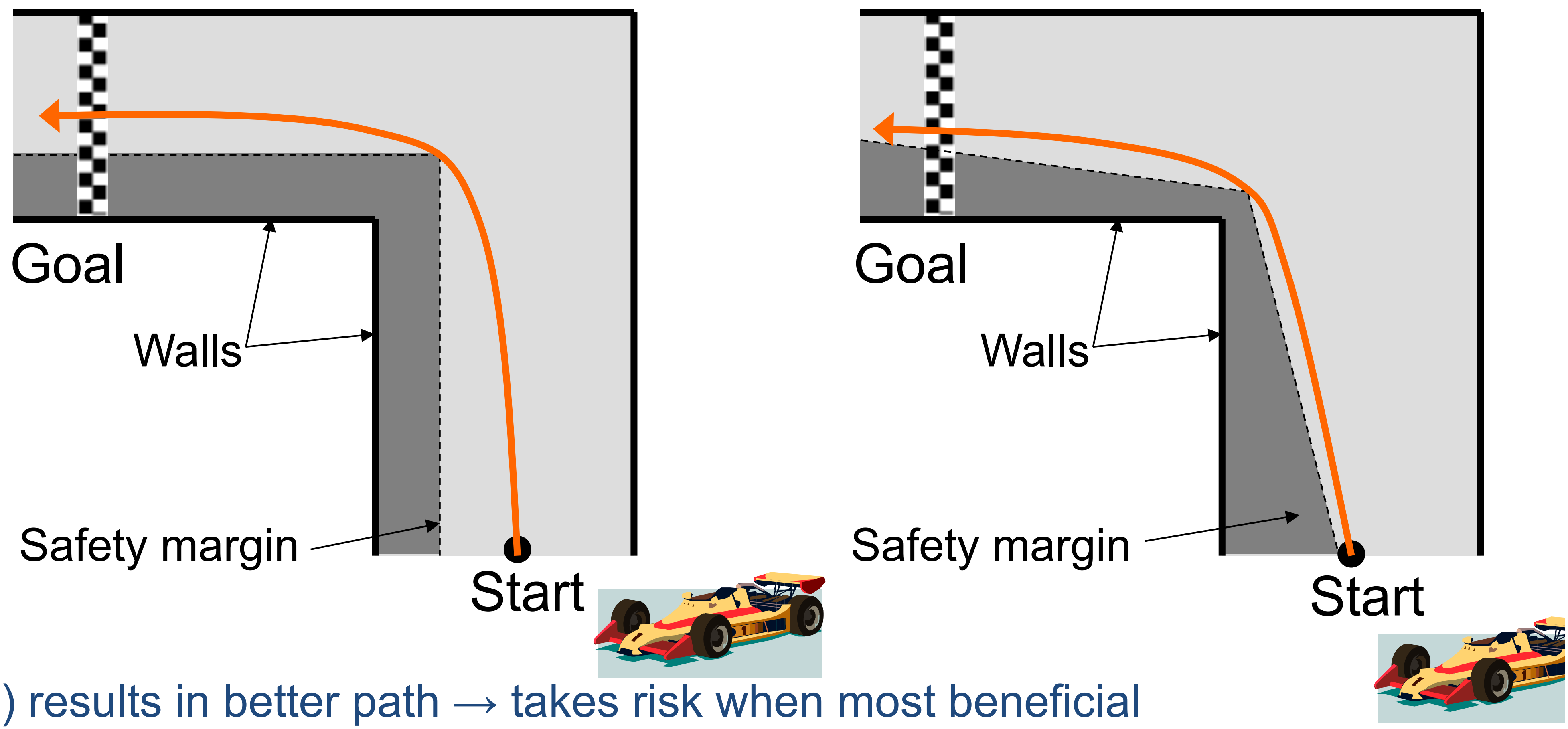
Risk bound

Idea:

1. Create safety margin that satisfies the risk bound from start to the goal.
2. Reduce to simpler, deterministic optimization problem.

Idea: Generate safety margin that satisfies risk bound while maximizing expected utility

(a) Uniform width safety margin **(b) Uneven width safety margin**



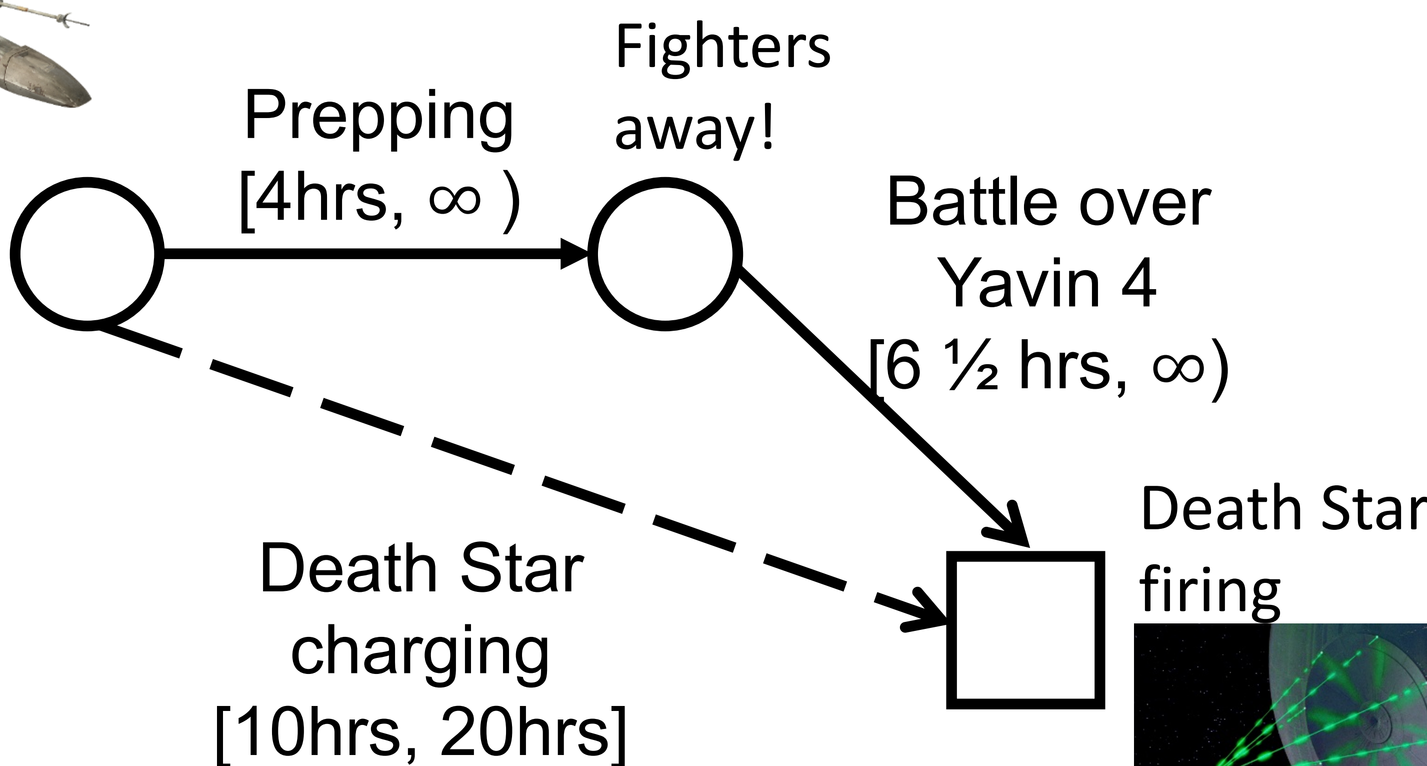
(b) results in better path → takes risk when most beneficial

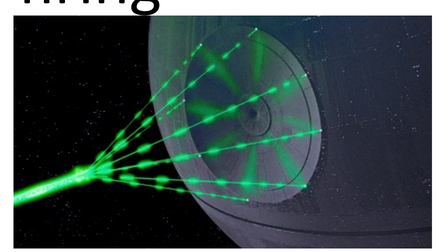
Approach: Algorithmic Risk Allocation

[Ono & Williams, AAAI 08]

Recall: Scheduling under Uncertainty

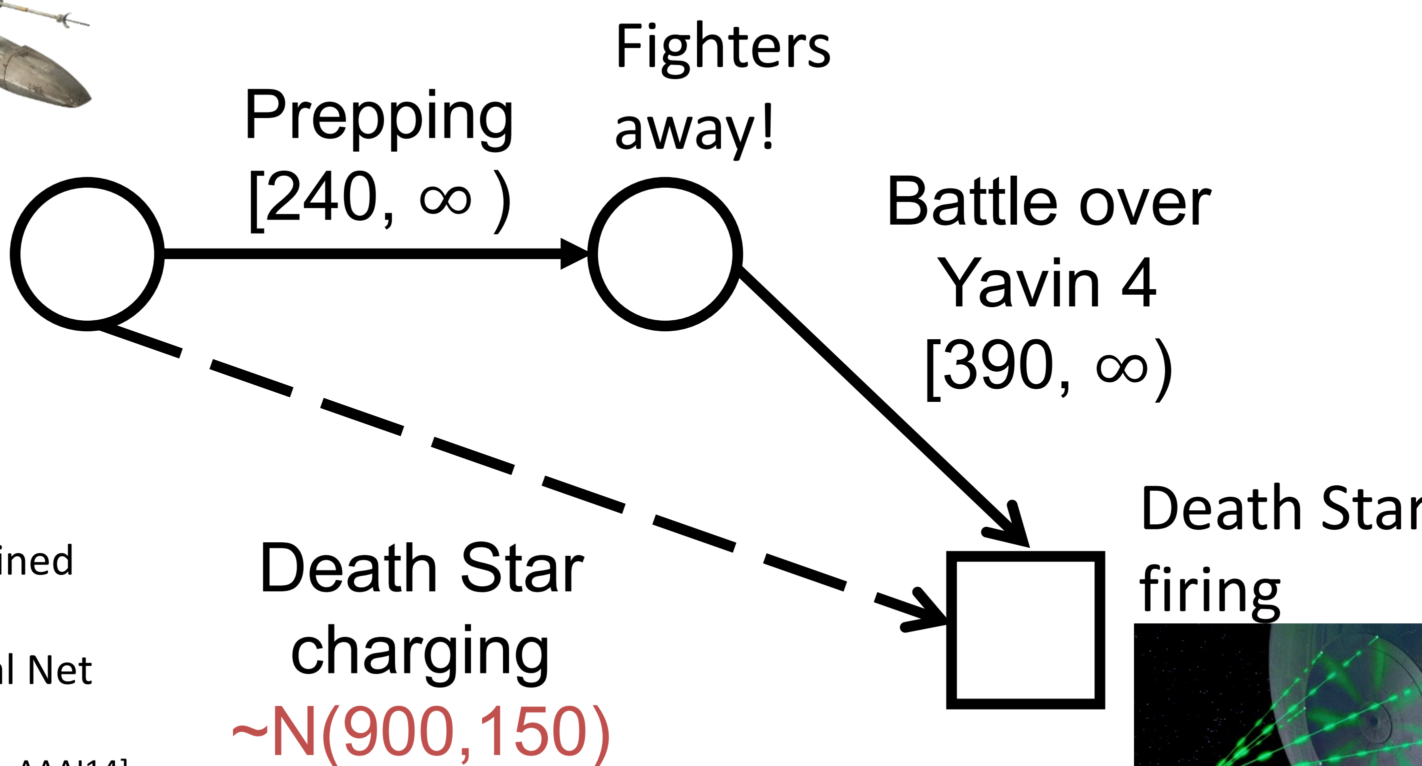
- That’s no moon – it’s a STNU!

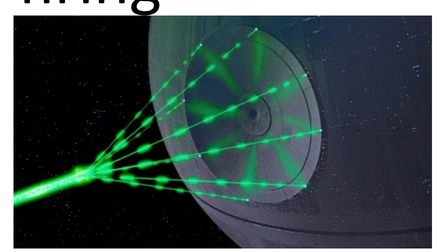




Risk-bounded Scheduling

- Find a schedule that works 95% of the time



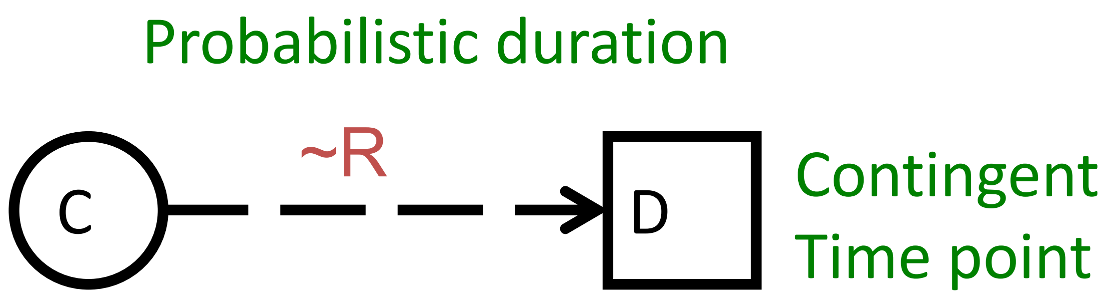


Chance-constrained
Probabilistic
Simple Temporal Net

[Fang, Yu, & Williams, AAAI14]

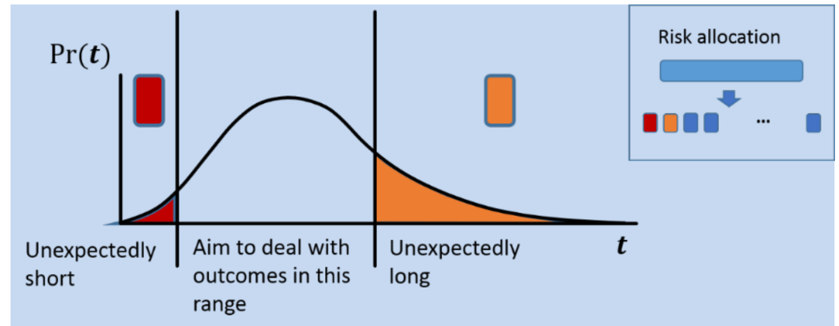
Probabilistic uncertainty

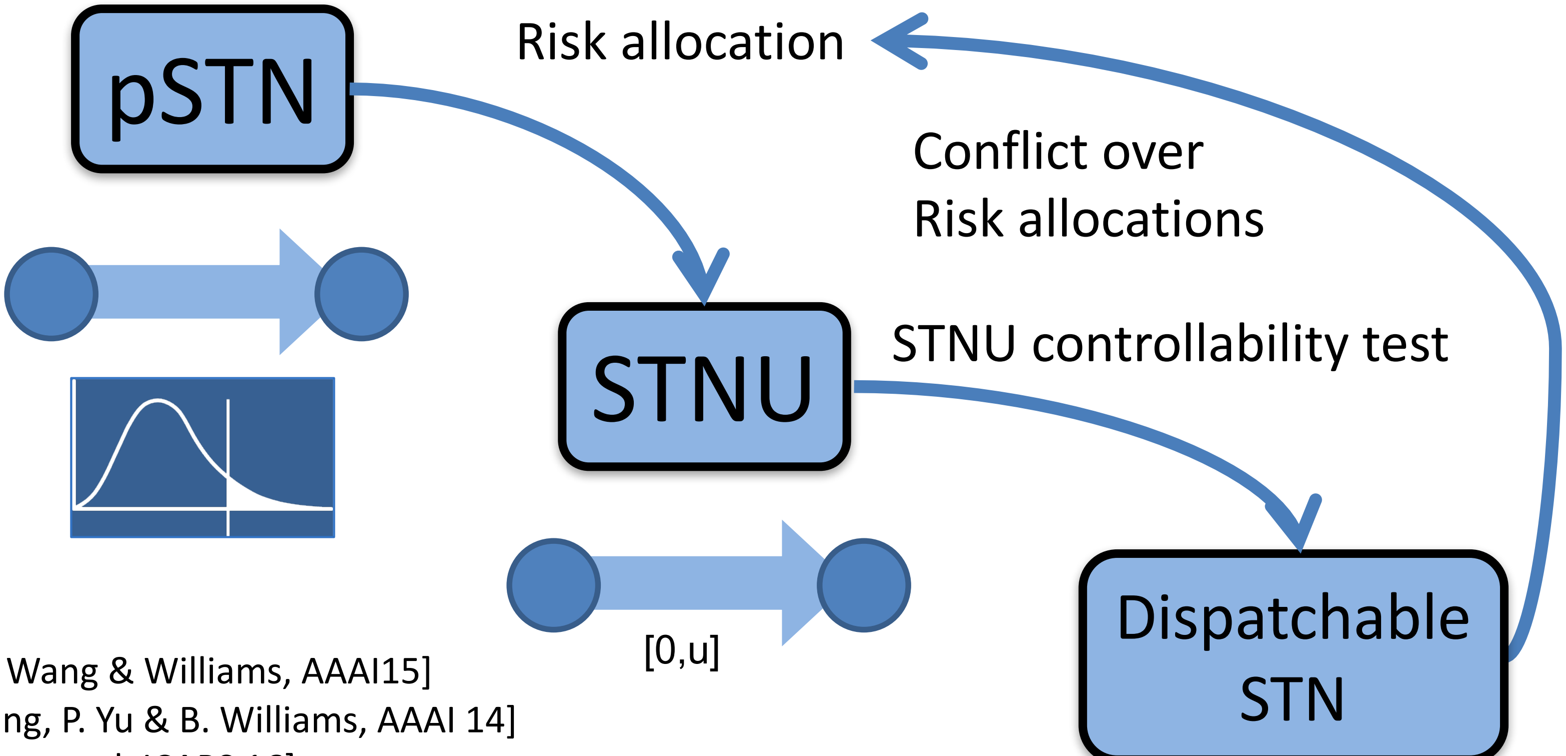
- probabilistic STN (pSTN):
  - Can say: The delay from C to D is described by random variable R



Ideas:

- Spend risk to ignore outcomes.
- Risk allocation maps problem to STNU
- When STNU is unsolvable,
  learn why risk allocation didn't work.





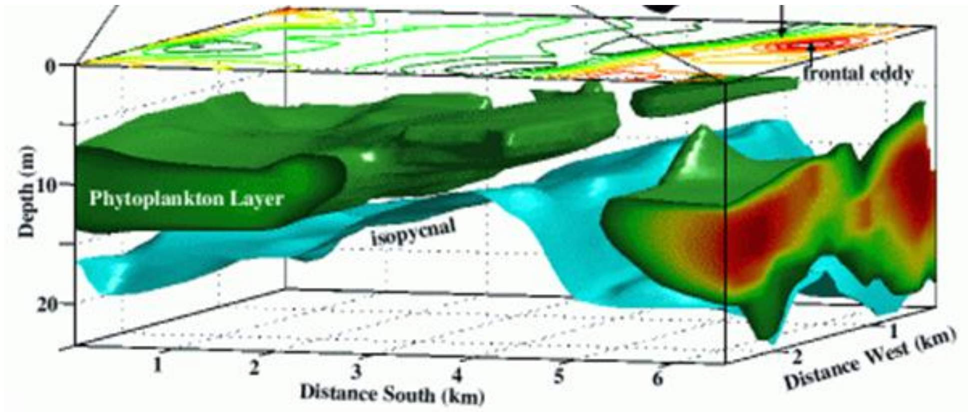
Rubato [A. Wang & Williams, AAAI15]
Picard [C. Fang, P. Yu & B. Williams, AAAI 14]
Paris [Santana, et al. ICAPS 16]

Outline

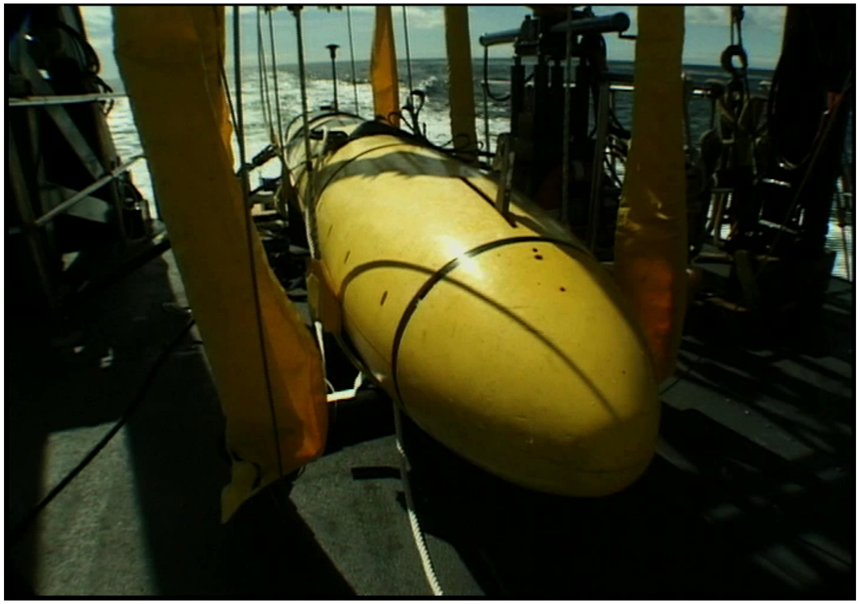
- A Risky Business
- Overview of Risk-bounded Planning
- Goal-directed Trajectory Planning
- Risk-bounded Trajectory Planning

Goal-directed Motion Planning

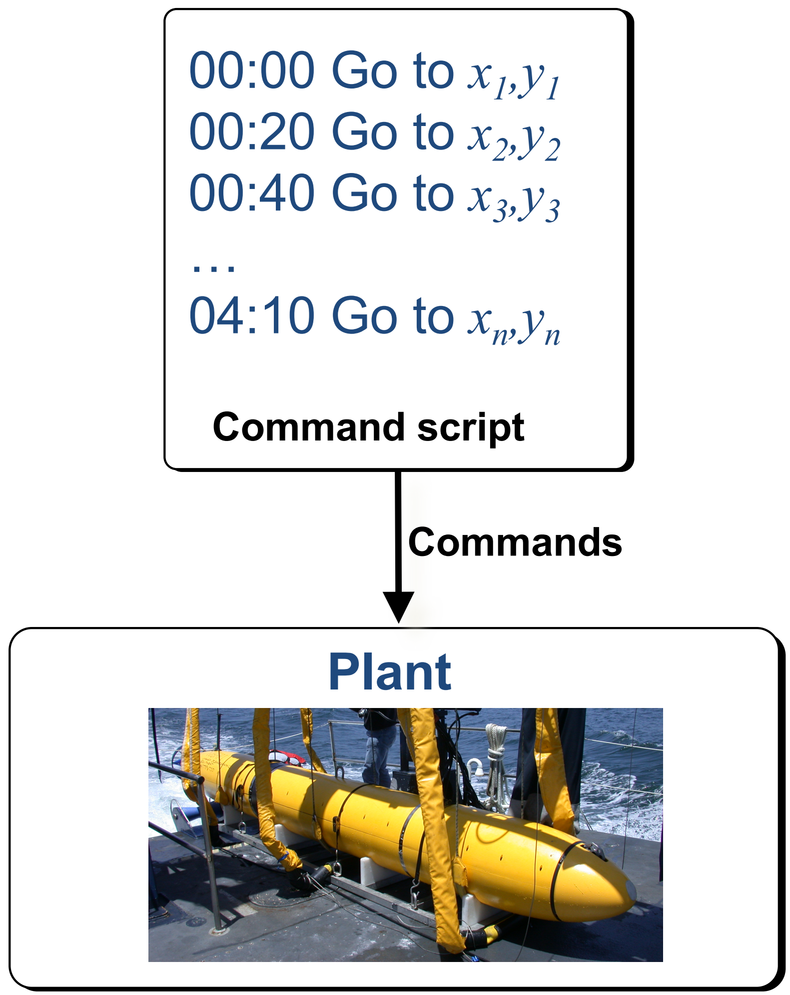
MBARI Dorado-class AUV:
- 6000m depth rated
- 20 hour operation
- Multi-beam sonar
- 3+ knots speed



Challenges:
- Long mission duration
- Limited communication
- GPS unavailable
- Uncertainty
  - tides and currents
  - estimation error



Dynamic Execution of Mission Scripts



Leaute & Williams, AAAI 05

Dynamic Execution of State Plans

"Explore bloom region for between 50 and 70 minutes. Afterwards, explore mapping region for between 40m and 50m. End in the pickup region. Avoid obstacles at all times. Complete the mission within 5hr"

State Plan

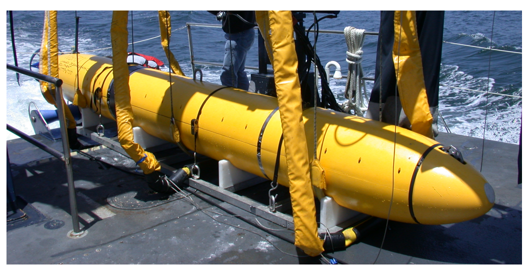
Sulu
Model-based Executive

Dynamics
+
Constraints

Observations

Commands

Plant

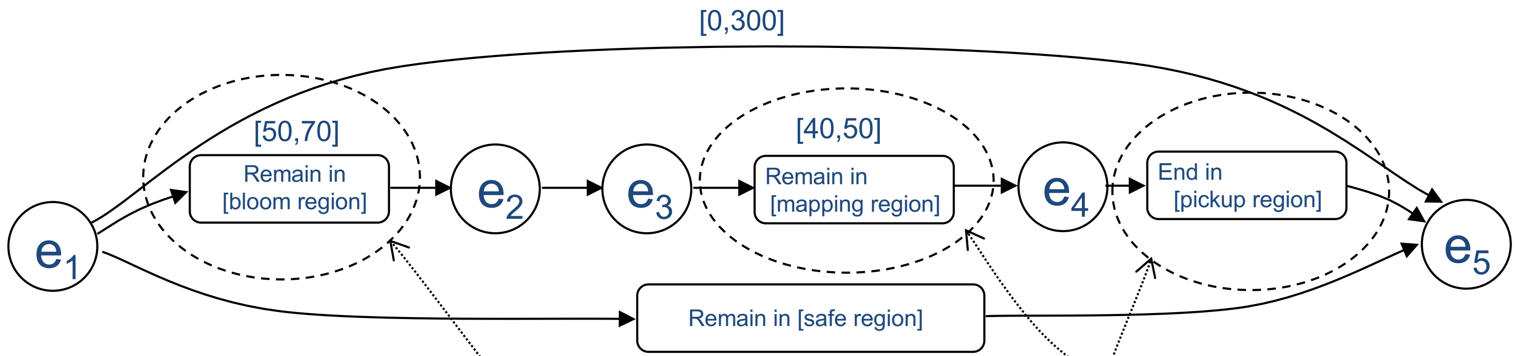


Optimal

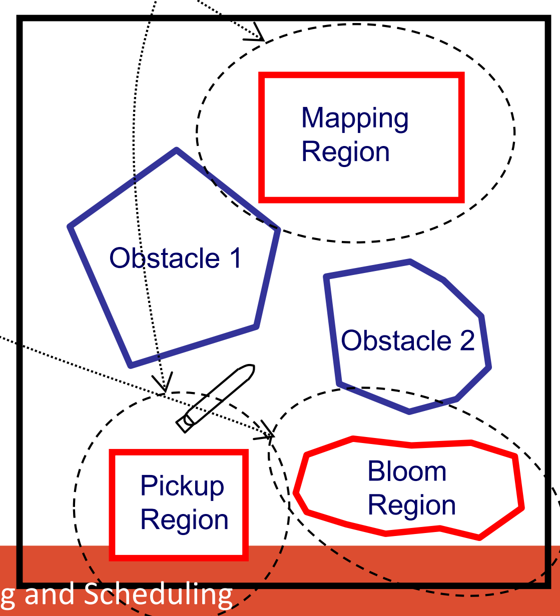
Leaute & Williams, AAAI 05

Sulu: Dynamic Execution of State Plans

A state plan is a model-based program that is unconditional, timed, and hybrid and provides flexibility in state and time.



"Explore bloom region for between 50 and 70 minutes. Afterwards, explore mapping region for between 40m and 50m. End in the pickup region. Avoid obstacles at all times. Complete the mission within 5hr"



Issue: Activities couple through time and state constraints.

Approach: Frame as Model-Predictive Control using Mixed Logic or Integer / Linear Programming.

[Leaute & Williams, AAAI 05]

State Plan specified as a RMPL Program

```
class MonterreyBayMission{
    [Dorado MapAuv…]
    method run(){
      [0, 5h] parallel {
        sequence {
          [50m,70m] MapAUV.RemainIn(bloom_region);
          [40m, 50m] MapAUV.RemainIn(map_region);
          MapAUV.EndIn (pickup_region)}
```

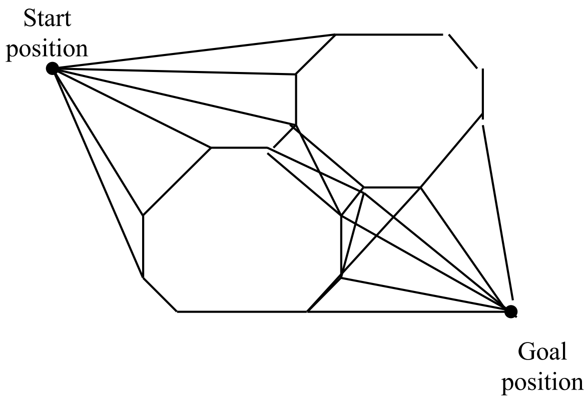
```
Program A ::=
   remain_in(R) | start_in(R) | end_in(R) |
   [lb, ub] A   |
   Sequence {A1; A2; ... } |
   Parallel {A1; A2; ... }
```

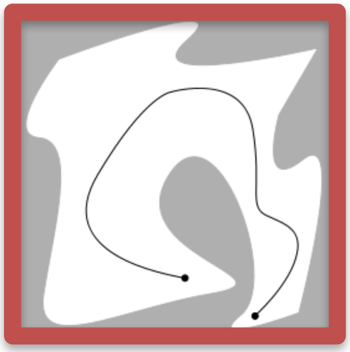
Motion Planning

Graph Search

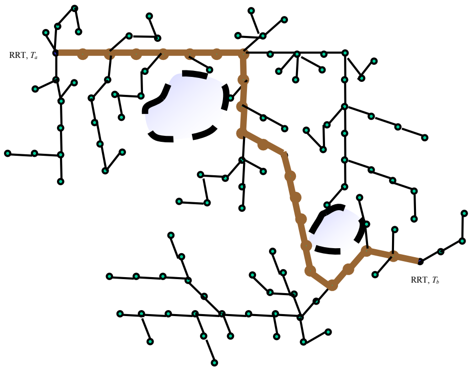
Optimization

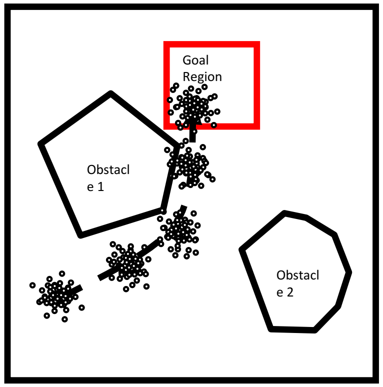
Exact





Sample-based





Goal-directed

Probabilistic

Risk-bounded

Input and Output

- **Sulu**: State Plan Motion Executive

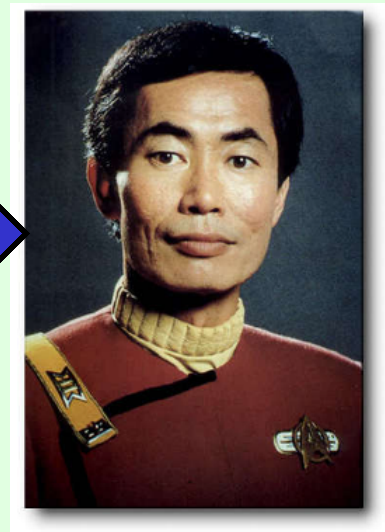
Input

Goal (QSP) +

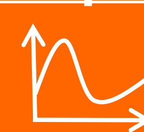
Dynamical Model

$(X_{t+1} = AX_t + Bu_t)$

Sulu



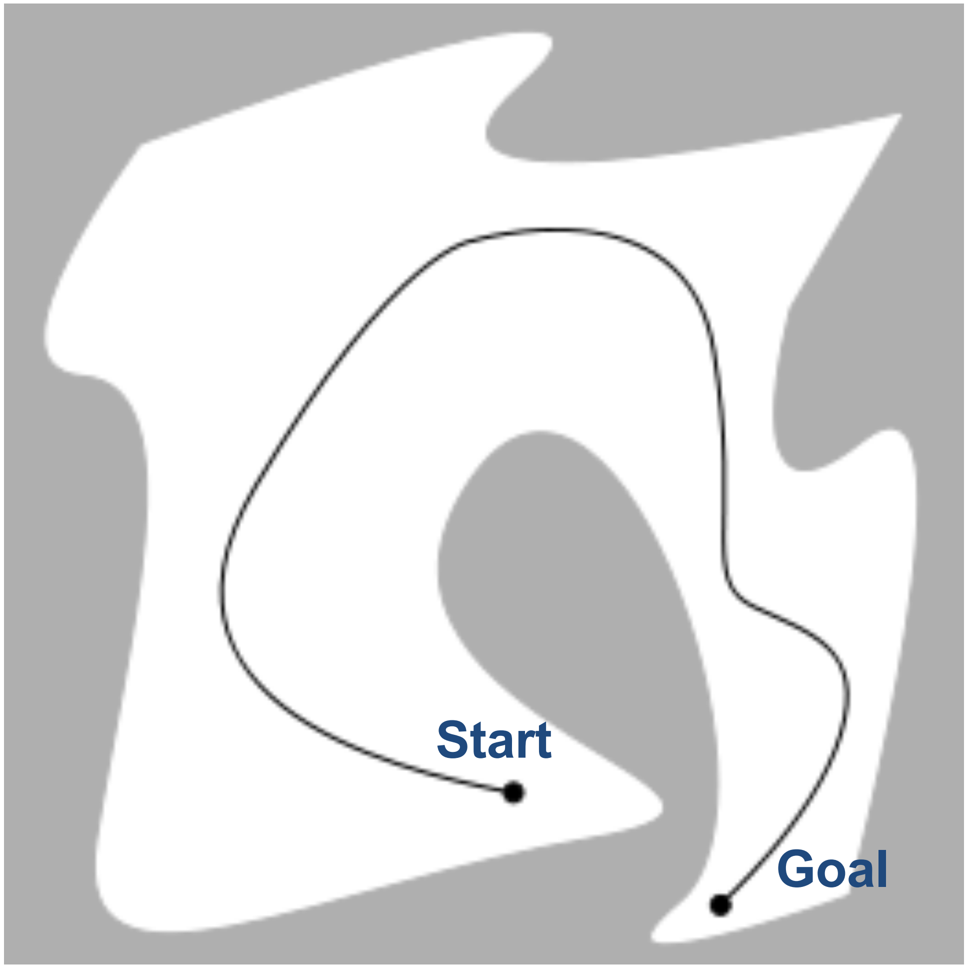
Output



Control Sequence

Constraint or Mathematical Programming Solver

Trajectory Optimization



- Plan control trajectory = constraint optimization

$$\min_{p} J(p)$$
$$s.t.$$
$$p \in P$$

p: path
P: Set of feasible paths
J: cost function

How do we encode the constraints for goal-directed trajectory optimization?

Finite Horizon Trajectory Optimization

How do we encode the constraints for goal-directed trajectory optimization?

- Formula … Program.

$$\min_{\mathbf{x}_{1:N},\mathbf{u}_{1:N}} J(\mathbf{x}_1 \cdots \mathbf{x}_N, \mathbf{u}_1 \cdots \mathbf{u}_N) \qquad \textbf{Cost function}$$

$$s.t.$$

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \quad (k = 0,1,\cdots N-1) \qquad \textbf{Dynamics}$$

$$\mathbf{H}\mathbf{x}_\mathbf{k} \le \mathbf{g} \quad (k = 0,1,\cdots N) \qquad \textbf{Spatial constraints}$$

$$\mathbf{x}_0 = \mathbf{x}_{\text{start}} \qquad \textbf{Initial position and velocity}$$

$$\mathbf{x}_N = \mathbf{x}_{\text{goal}} \qquad \textbf{Goal position and velocity}$$

$$-\mathbf{u}_{\max} \le \mathbf{u}_k \le \mathbf{u}_{\max} \quad (k = 0,1,\cdots N-1) \qquad \textbf{Actuation limits}$$

$$\mathbf{x}_k \equiv \begin{pmatrix} x_k & y_k & \dot{x}_k & \dot{y}_k \end{pmatrix}^T, \ \mathbf{u}_k \equiv \begin{pmatrix} F_{x,k} & F_{y,k} \end{pmatrix}^T$$

Encoding Dynamics and Actuation Constraints

- 2-D Omni-dimensional Holonomic Vehicle in a room

Dynamics

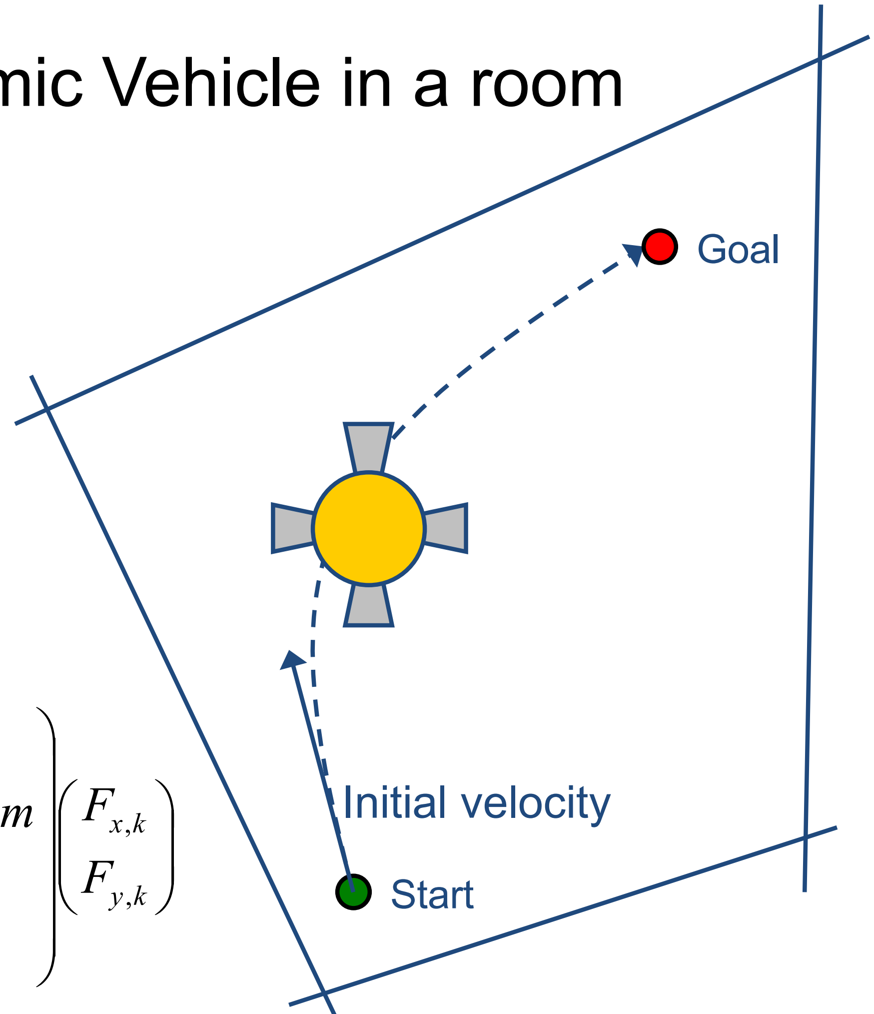
$$m\begin{pmatrix}\ddot{x}\\ \ddot{y}\end{pmatrix}=\begin{pmatrix}F_x\\ F_y\end{pmatrix}$$

$$|F_x|\le F_{\max}, |F_y|\le F_{\max}$$ (Thrust limits)

Discrete-time dynamics*
(zero-order hold assumption)

$$\begin{pmatrix}x_{k+1}\\ y_{k+1}\\ \dot{x}_{k+1}\\ \dot{y}_{k+1}\end{pmatrix}=\begin{pmatrix}1 & 0 & \Delta t & 0\\ 0 & 1 & 0 & \Delta t\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix}x_k\\ y_k\\ \dot{x}_k\\ \dot{y}_k\end{pmatrix}+\begin{pmatrix}0.5\Delta t^2/m & 0\\ 0 & 0.5\Delta t^2/m\\ \Delta t/m & 0\\ 0 & \Delta t/m\end{pmatrix}\begin{pmatrix}F_{x,k}\\ F_{y,k}\end{pmatrix}$$

$$\boxed{x_{t+1} = \quad Ax_t \quad + \quad Bu_t}$$



***How to obtain discrete-time dynamics from continuous-time dynamics?**
- Take a look at control theory text books (chapter on discrete-time system)
- Use MATLAB **c2d** command

Encoding Spatial Constraints

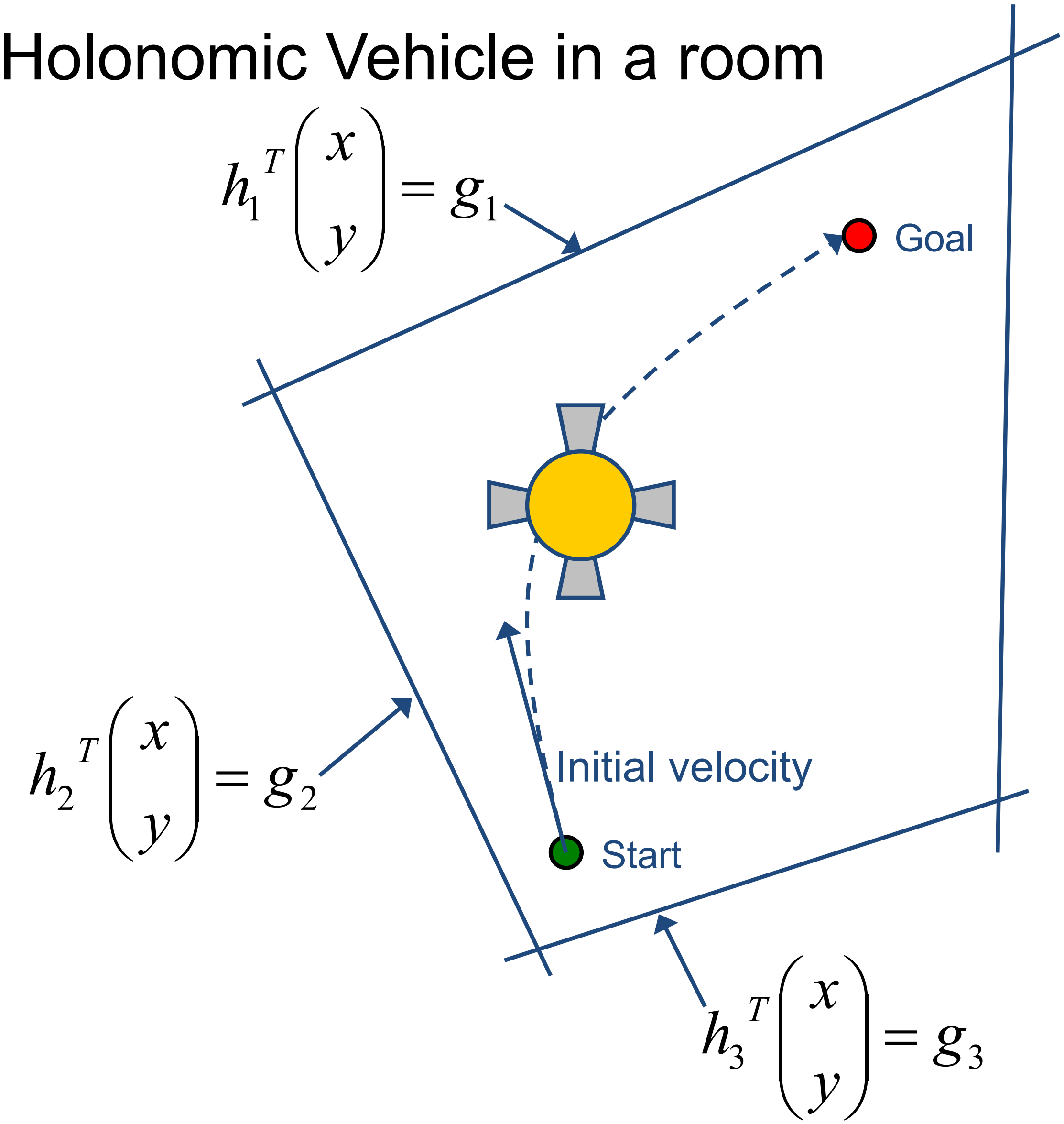
- 2-D Omni-dimensional Holonomic Vehicle in a room

Spatial constraints:
Vehicle must be in the room

$$\bigwedge_{n=1}^{4} h_n^T \begin{pmatrix} x \\ y \end{pmatrix} \le g_n$$

or

$$\mathbf{Hx} \le \mathbf{g}$$



Encoding Cost

- What cost function should we use?
  - Example: minimum control effort

$$J(\mathbf{x}_1 \cdots \mathbf{x}_N, \mathbf{u}_1 \cdots \mathbf{u}_{N-1}) = \sum_{k=1}^{N-1} \begin{pmatrix} 1 & 1 \end{pmatrix} |\mathbf{u}_k| = \sum_{k=1}^{N-1} |F_{x,k}| + |F_{y,k}|$$

  - Problem: This is not a linear function!!
  - There are tricks.

$$\min |u| \Leftrightarrow \begin{array}{l} \min u^+ + u^- \\ u = u^+ - u^- \\ u^+ \geq 0, u^- \geq 0, \end{array}$$

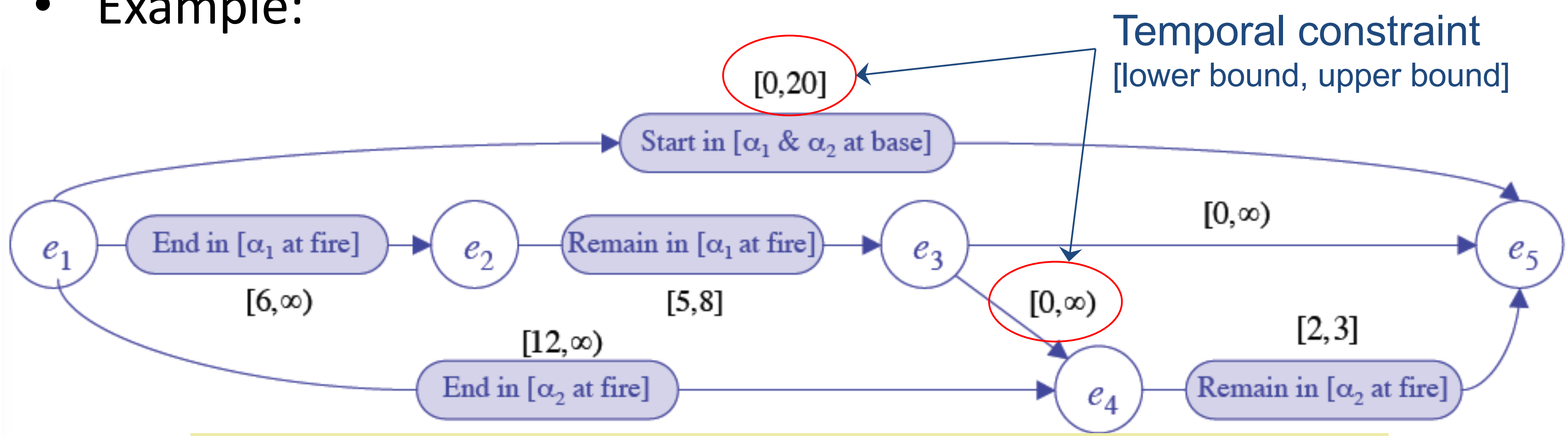
or

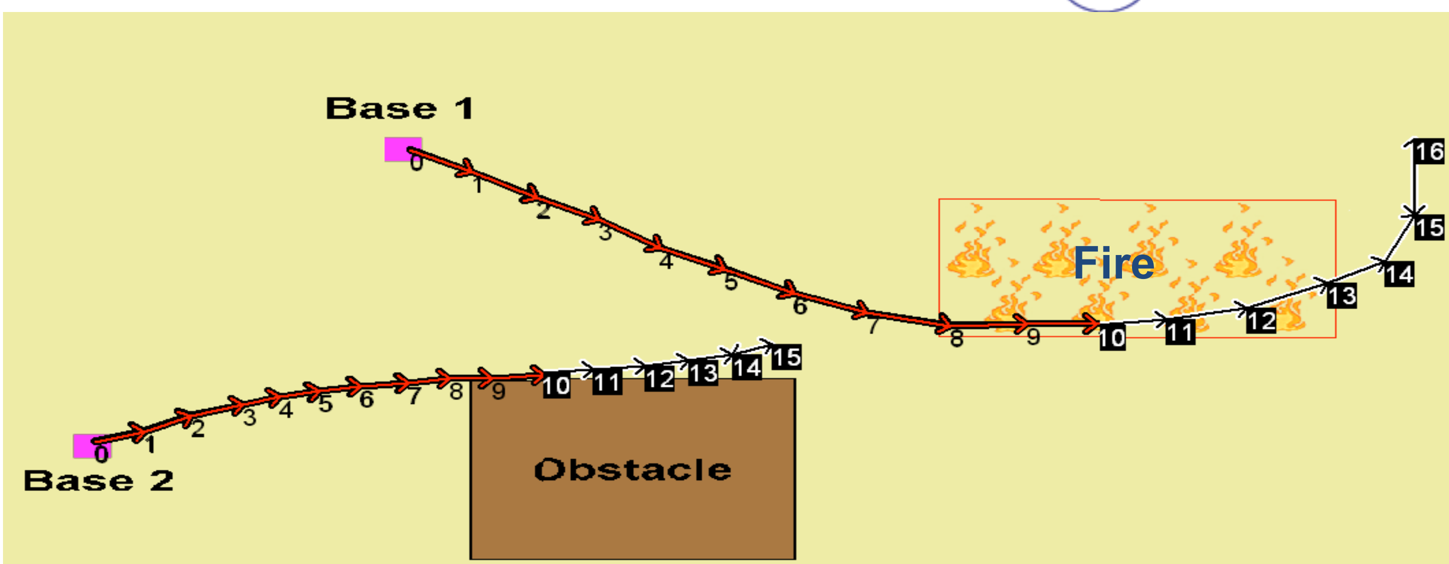
$$\begin{array}{l} \min v \\ v \geq u, v \geq -u, \end{array}$$

Encoding Qualitative State Plans

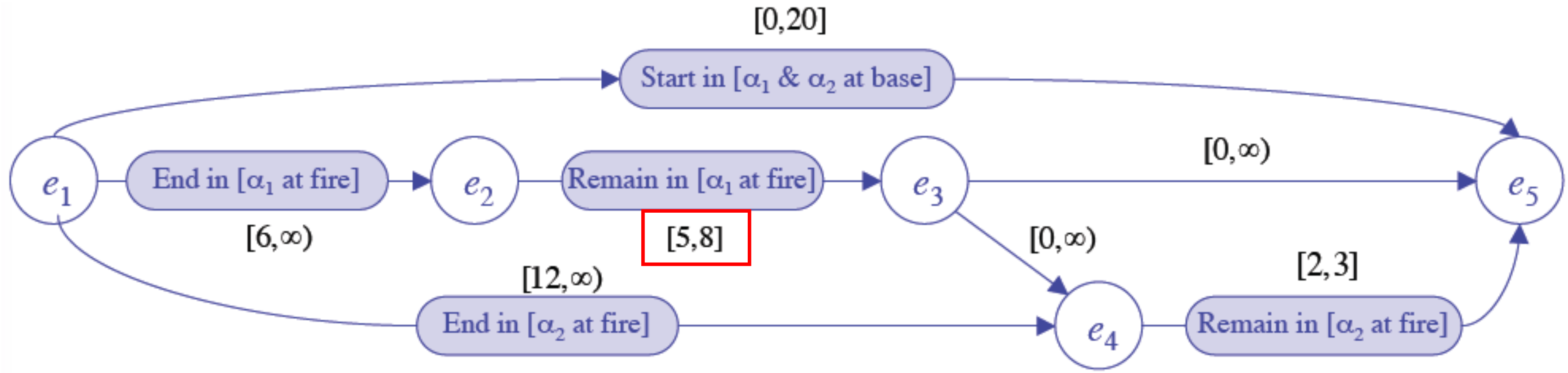
Sulu [Leaute & Williams, AAAI05]

- Example:





Encoding Temporal Constraints

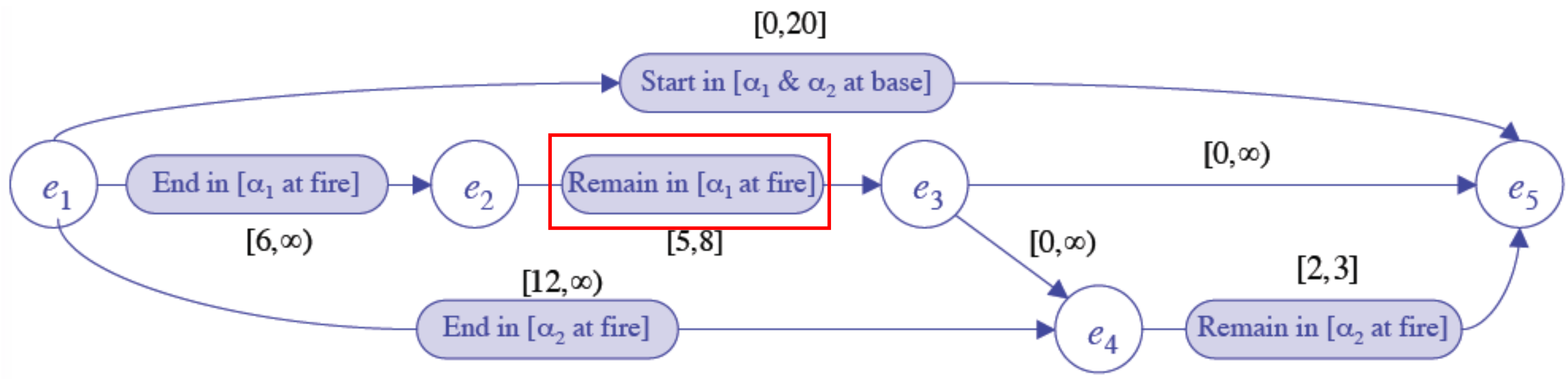


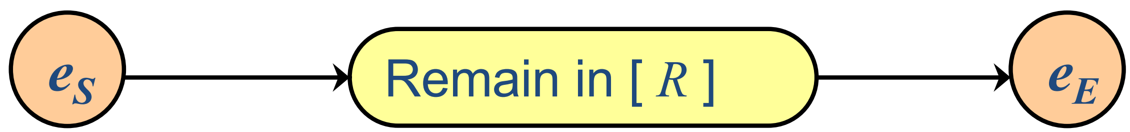
$$LB \leq T(e_E) - T(e_S) \leq UB$$

$T(e_E), T(e_S)$: decision variables

•Thomas Léauté, "*Coordinating Agile Systems through the Model-based Execution of Temporal Plans*, " S. M. Thesis, Massachusetts Institute of Technology, August 2005.
•Thomas Léauté, Brian Williams, "Coordinating Agile Systems Through the Model-based Execution of Temporal Plans," *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05),* Pittsburgh, PA, July 2005, pp. 114-120.

Encoding "Remain In" Constraints

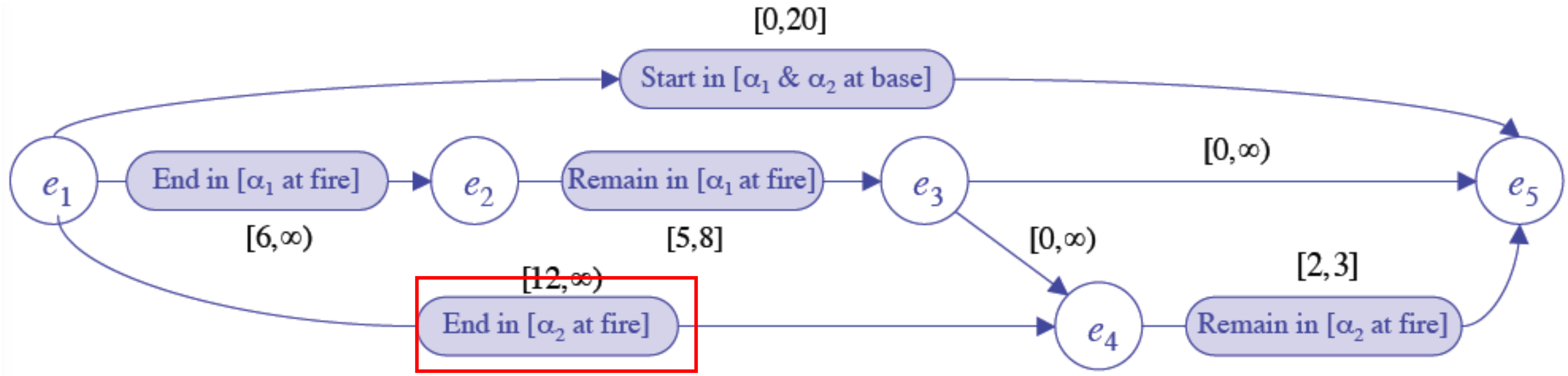




$$\bigwedge_{k=0}^{k=N} \left\{ T(e_S) \le t_k \le T(e_E) \boxed{\Rightarrow} \mathbf{x}_k \in R \right\}$$

•Thomas Léauté, "*Coordinating Agile Systems through the Model-based Execution of Temporal Plans*, " S. M. Thesis, Massachusetts Institute of Technology, August 2005.
•Thomas Léauté, Brian Williams, "Coordinating Agile Systems Through the Model-based Execution of Temporal Plans," *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05),* Pittsburgh, PA, July 2005, pp. 114-120.

Encoding “End In” Constraints

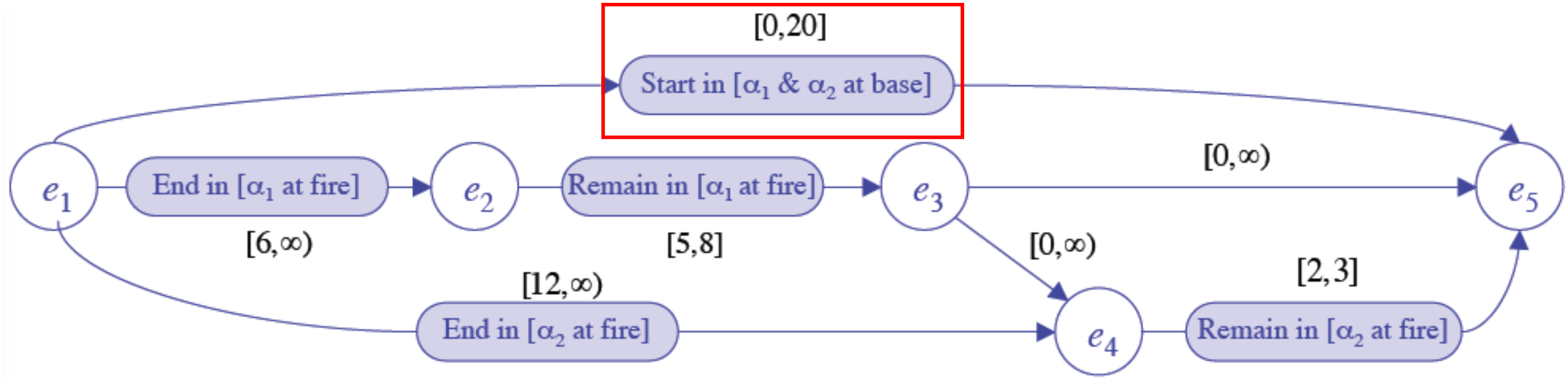


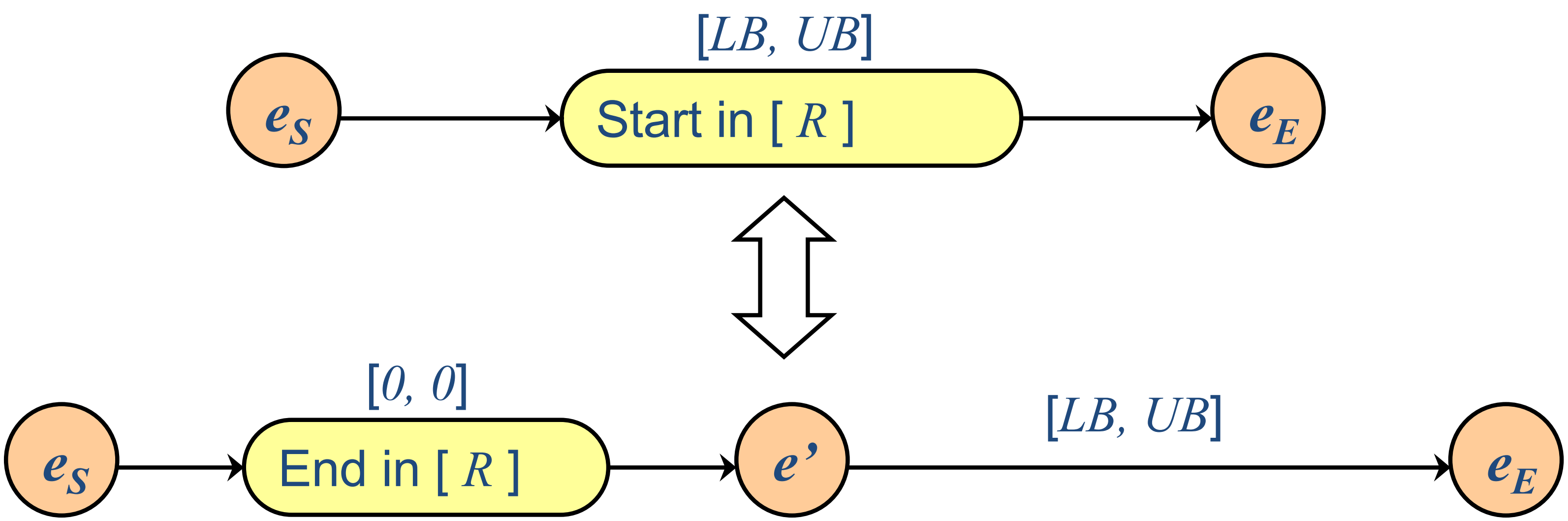
Variable

$$\bigvee_{k=0}^{k=N} \left\{ t_k - \frac{\Delta t}{2} \le T(e_E) \le t_k + \frac{\Delta t}{2} \wedge \mathbf{x}_k \in R \right\}$$

Constant

Encoding “Start In” Constraints



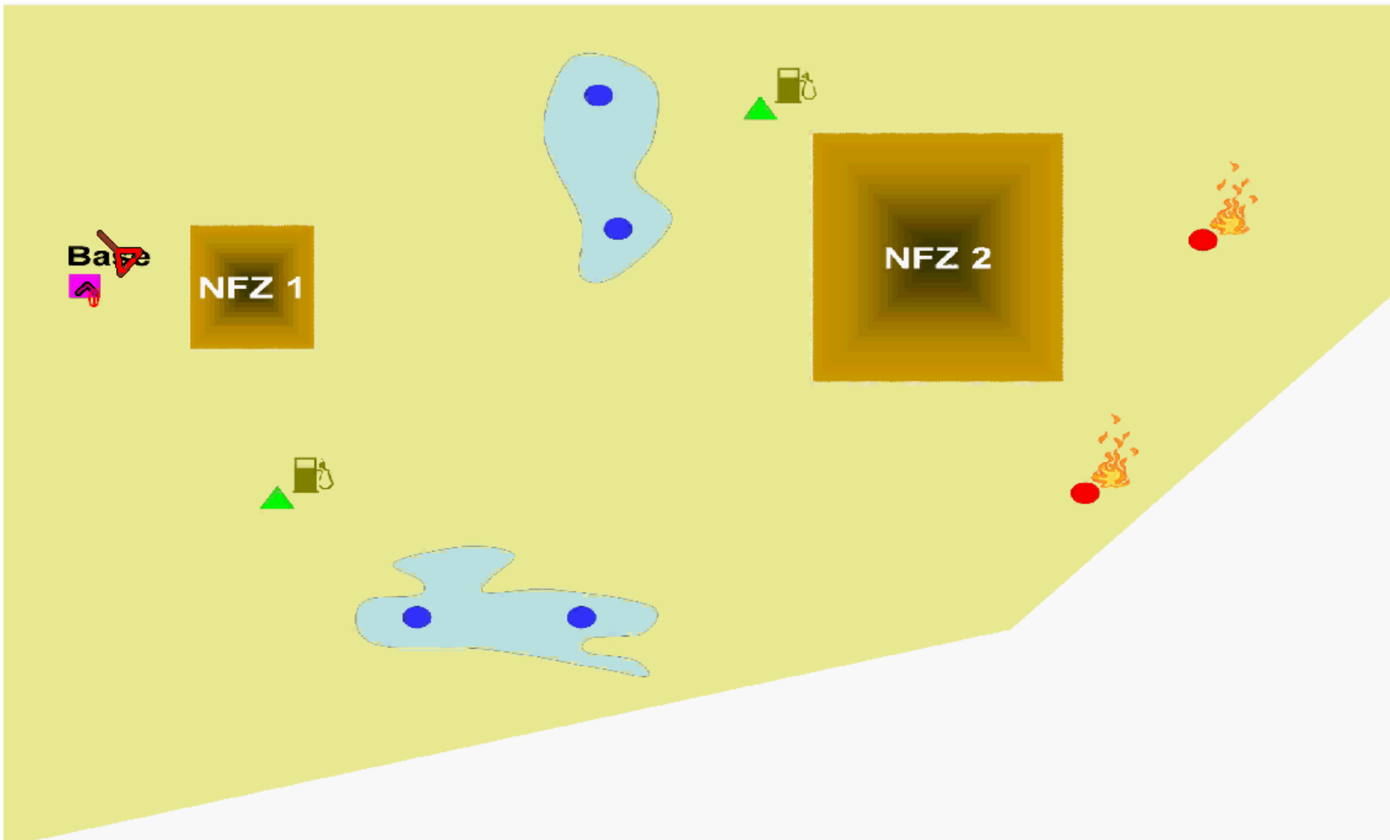


•Thomas Léauté, "*Coordinating Agile Systems through the Model-based Execution of Temporal Plans*, " S. M. Thesis, Massachusetts Institute of Technology, August 2005.
•Thomas Léauté, Brian Williams, “Coordinating Agile Systems Through the Model-based Execution of Temporal Plans,” *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05),* Pittsburgh, PA, July 2005, pp. 114-120.

Goal-directed Receding Horizon Control

Fire Fighter states goals:

"Need fires out at Locations 1 & 2
and back to Base within an hour."



Location 1

Location 2

1. Activity Planner plans activities, route and schedule $\Rightarrow$ QSP.
2. QSP executed by goal-directed receding horizon trajectory planner

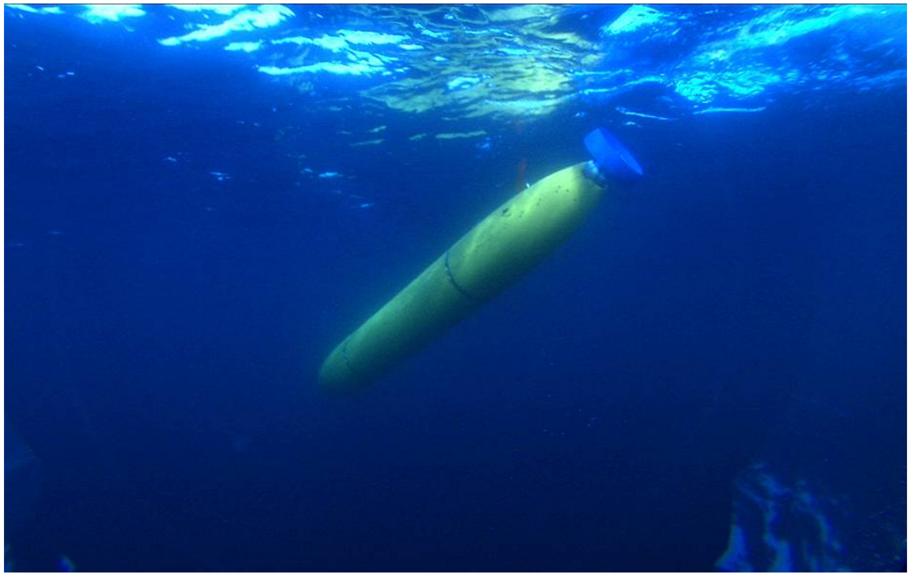
Key takeaways

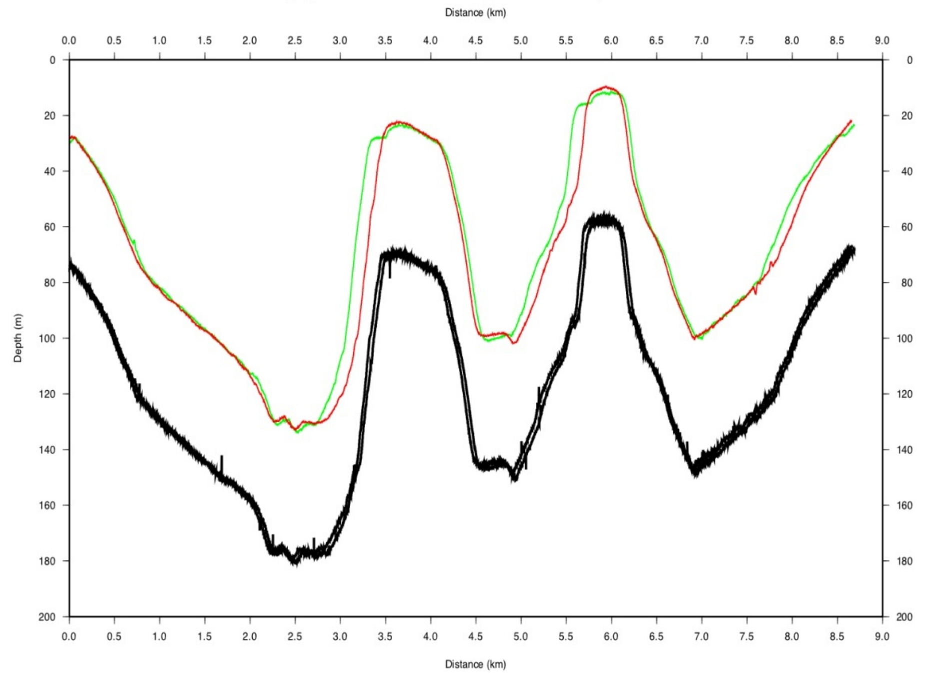
- Motions need to be generated in light of higher level goals.
- For under actuated system, activities and motions couple through state and temporal constraints.

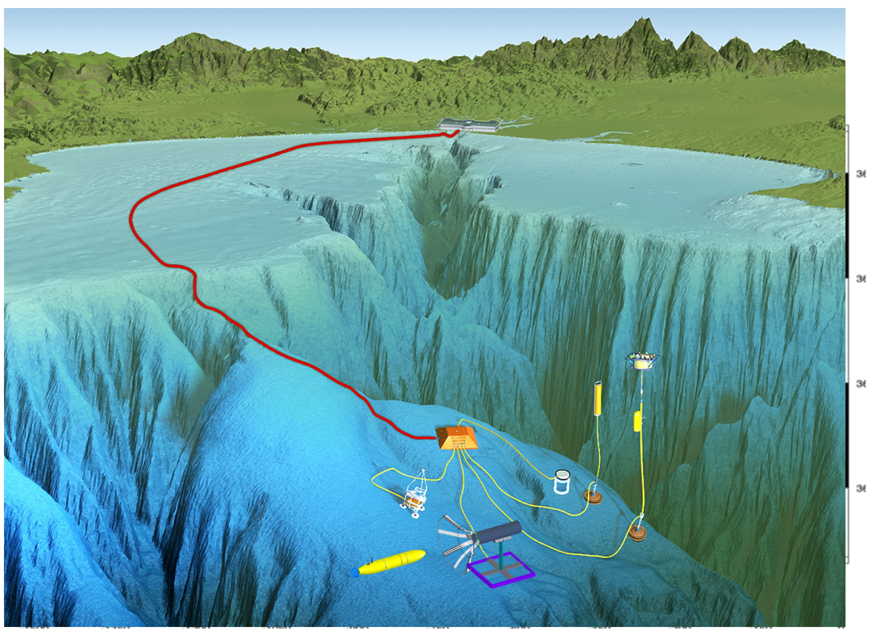
Outline

- A Risky Business
- Overview of Risk-bounded Planning
- Goal-directed Trajectory Planning
- Risk-bounded Trajectory Planning

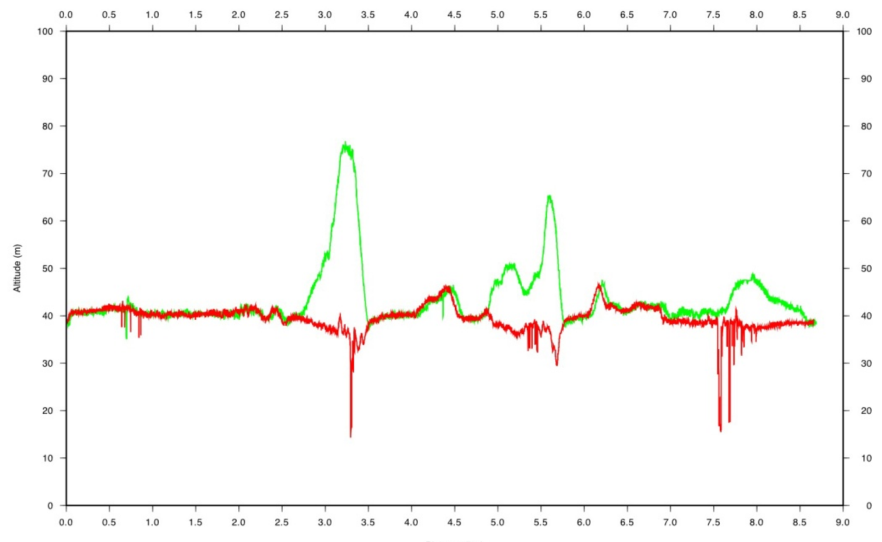
Depth Navigation for Bathymetric Mapping – Jan. 23rd, 2008





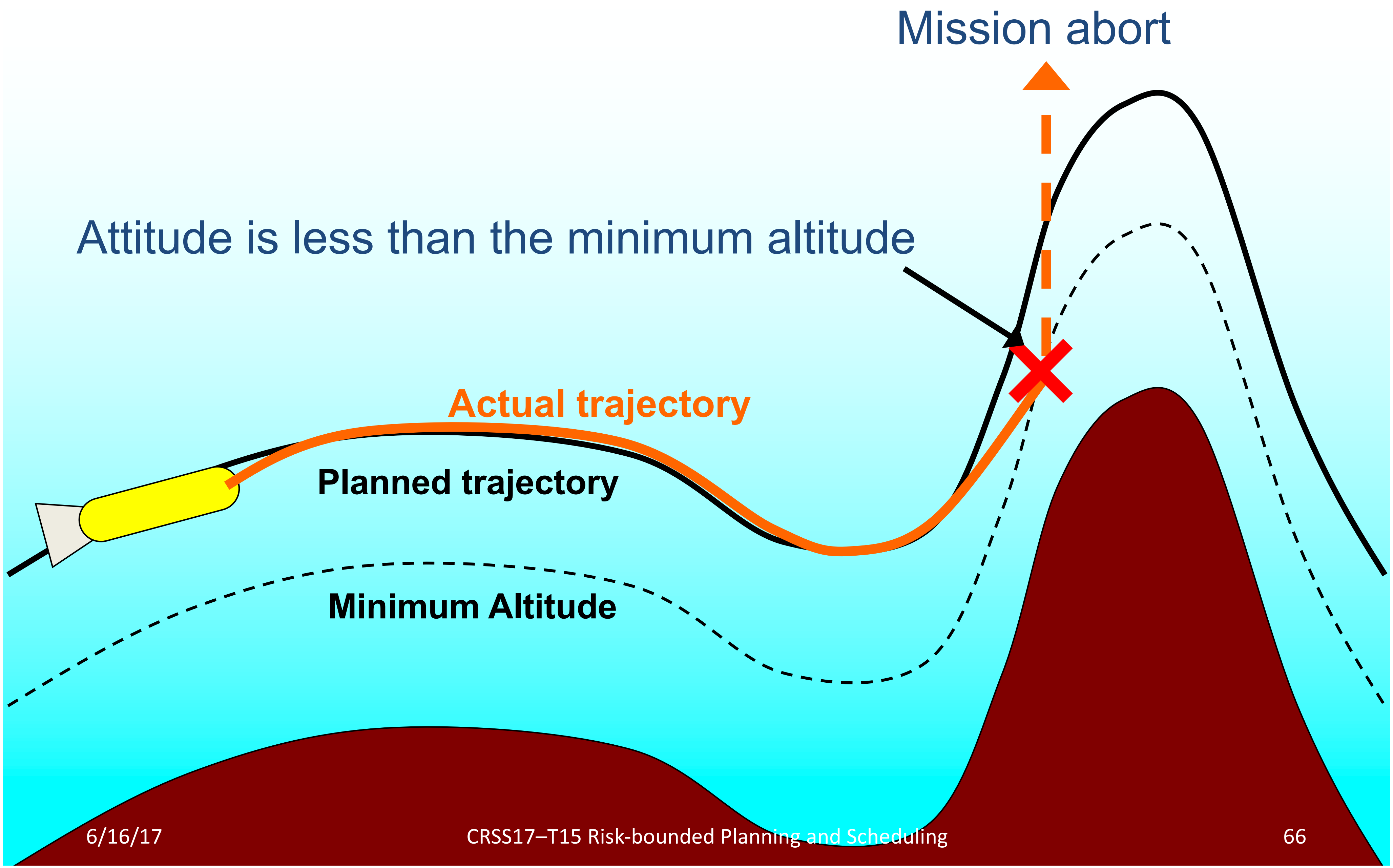


Courtesy MBARI




>$1M

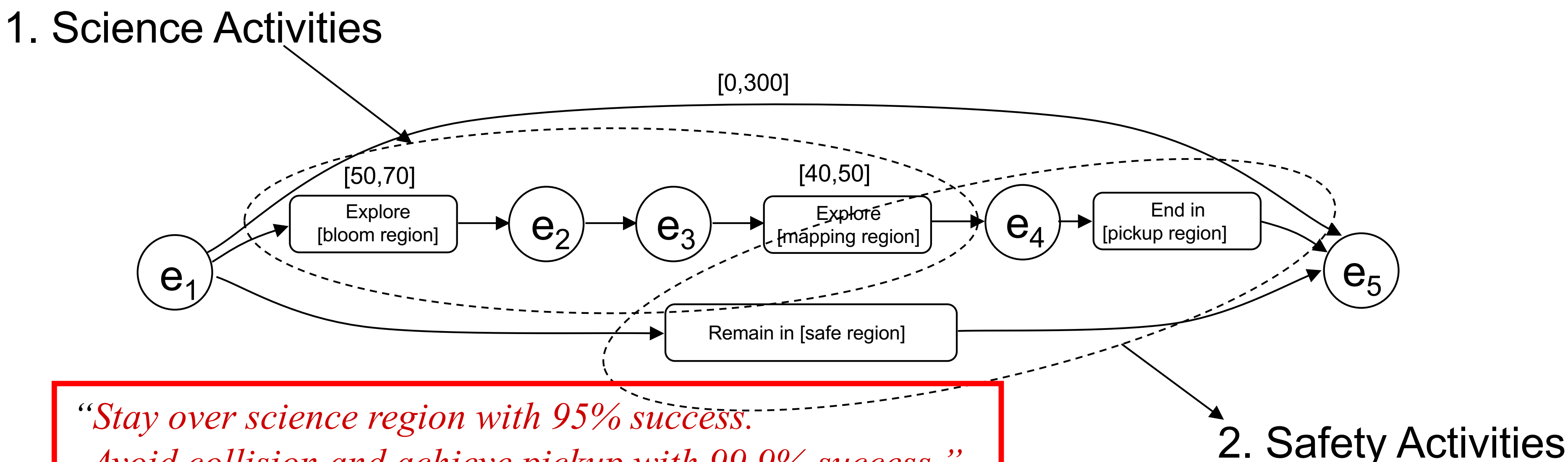
Issue: Frequent Mission Aborts



Input Goal: Risk-bounded State Plan

Operator: Specifies acceptable risk.

Executive: Decides how to use risk effectively.



"Stay over science region with 95% success.
Avoid collision and achieve pickup with 99.9% success."

Constraints on risk of failure (Chance Constraints):

1. *p(* Remain in [bloom region] *fails* OR Remain in [mapping region] *fails*) < 5%.
2. *p(* End in [pickup region] *fails* OR Remain in [safe region] *fails*) < .1%.

Instance of Chance-constrained Programming.

Input Goal: Risk-bounded Program (cRMPL)

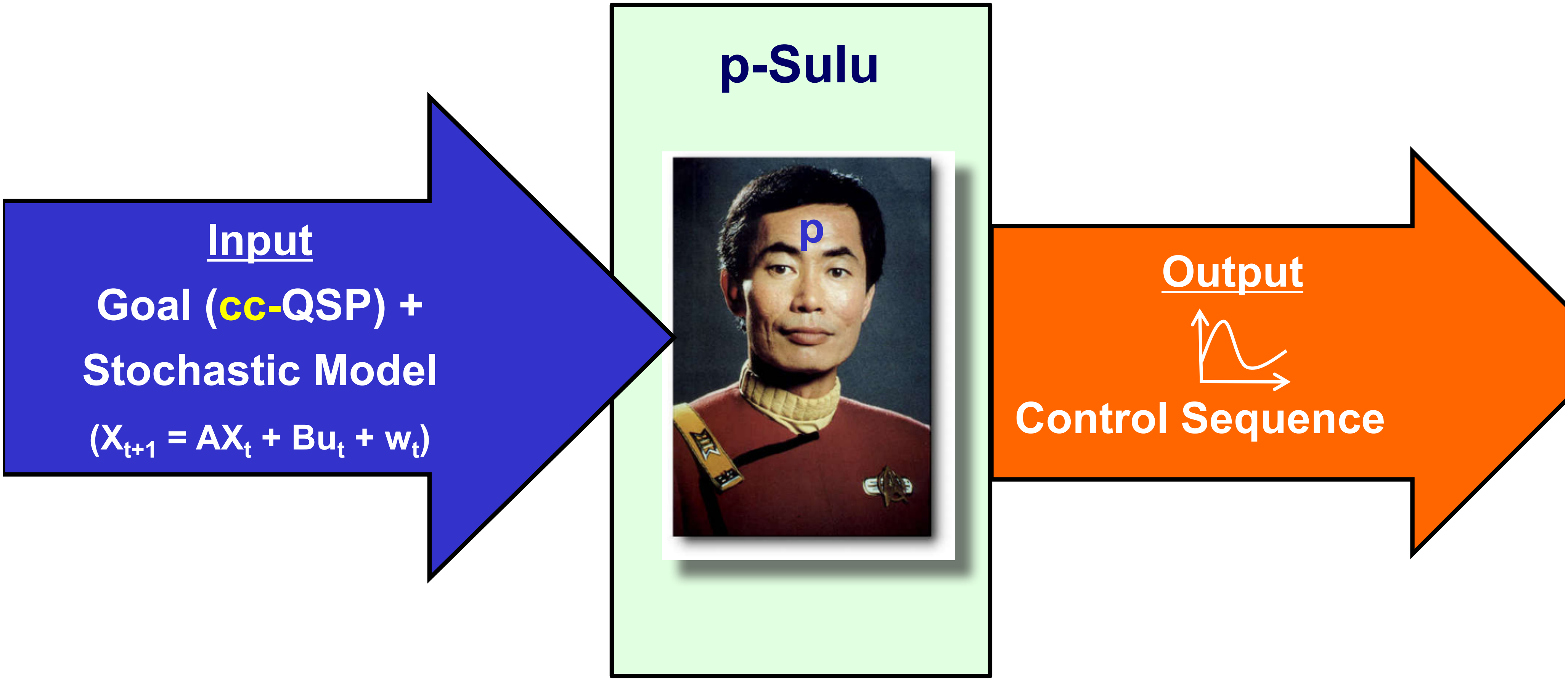
```
class MonterreyBayMission{
   [Dorado MapAuv…]
   method run(){
     [0, 5h] parallel {
       chance_constraint: 0.95 sequence {
         [50m,70m] MapAUV.RemainIn(bloom_region);
         [40m, 50m] MapAUV.RemainIn(map_region);}

       chance_constraint: 0.99.9 [0, 5h] parallel {
         MapAUV.RemainOutside (sea_floor);
         MapAUV.EndIn (pickup_region);}}}
```

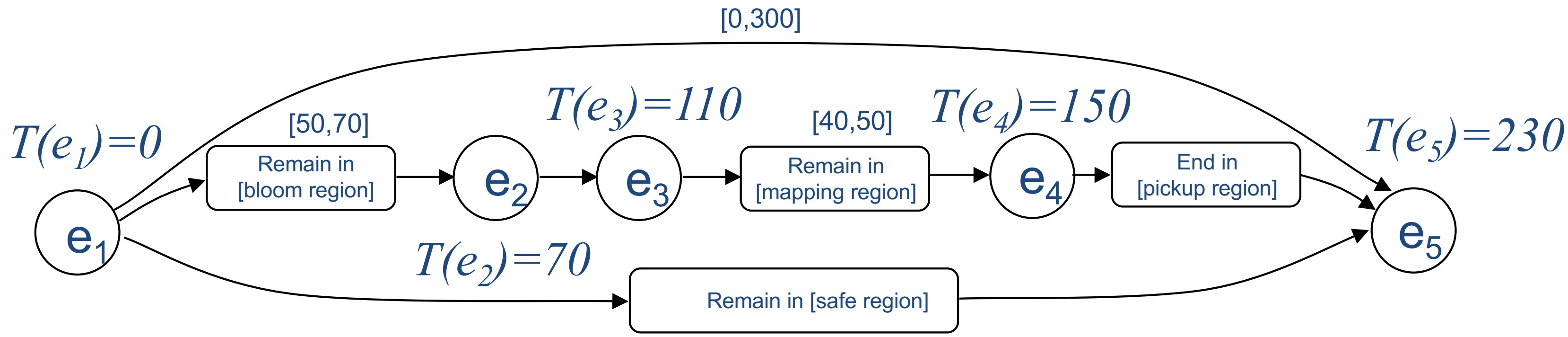
```
Program A ::=
   remain_in(R) | start_in(R) | end_in(R) |
   [lb, ub] A   |
   chance_constraint: Delta A  |
   Sequence {A1; A2; ... } |
   Parallel {A1; A2; ... }
```

Input and Output

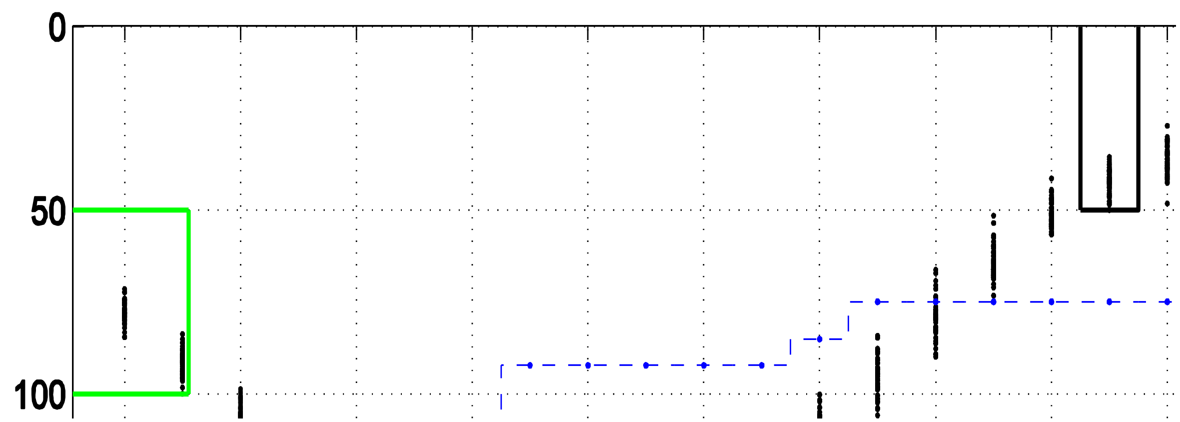
- **p-Sulu**: Probabilistic Sulu (State Plan Executive)

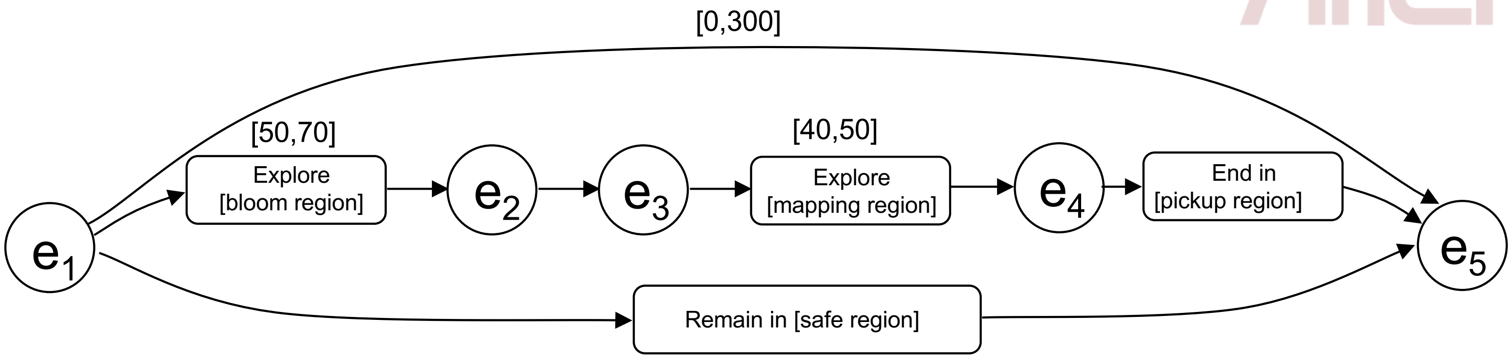


Output: Simulation of Control Sequence



"Stay over science region with 95% success.
Avoid collision and achieve pickup with 99.9% success."

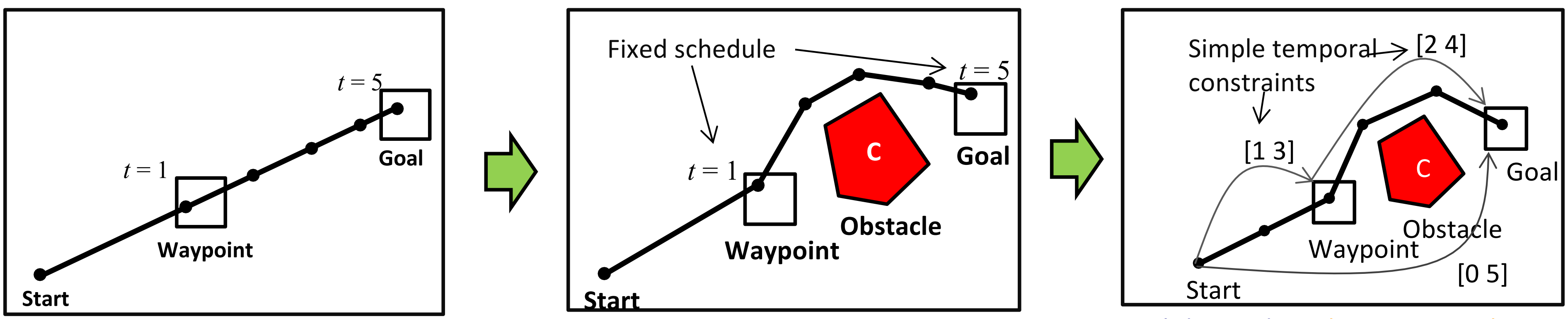




> *"Stay over science region with 95% success.*
> *Avoid collision and achieve pickup with 99.9% success."*

Approach: Solve increasingly expressive problems

(QSP)



Convex chance-constrained traj opt

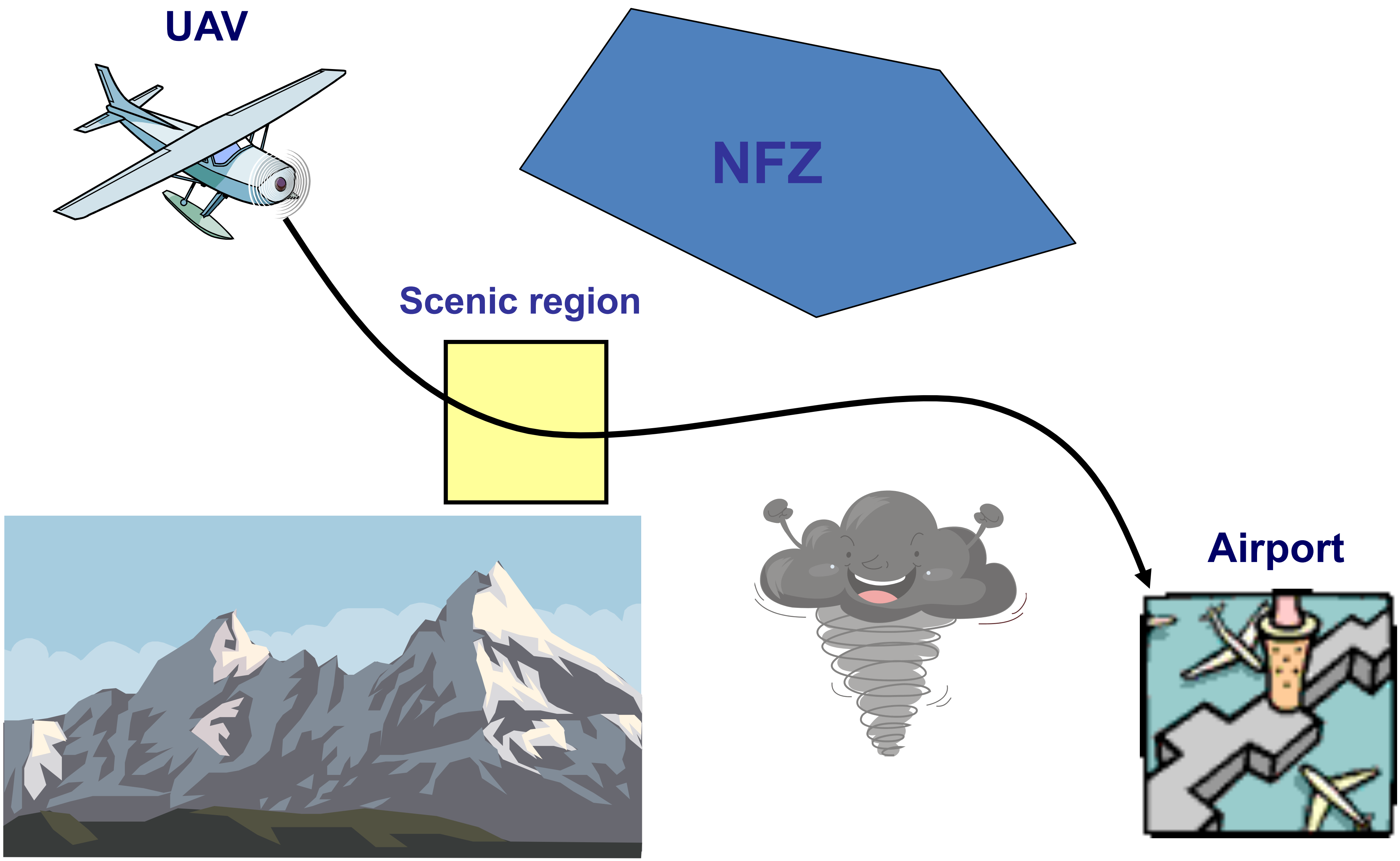
Non-convex, chance-constrained traj opt

Goal-directed qualitative state plan traj opt

Outline

- A Risky Business
- Overview of Risk-bounded Planning
- Goal-directed Trajectory Planning
- Risk-bounded Trajectory Planning
  - Problem Statement
  - Elliptic Approximation
  - Iterative Risk Allocation (IRA)
  - Convex Risk Allocation (CRA)

What are the uncertainties and risks?



Incorporating Uncertainty

- Deterministic discrete-time LTI model.

$$x_{t+1} = Ax_t + Bu_t$$

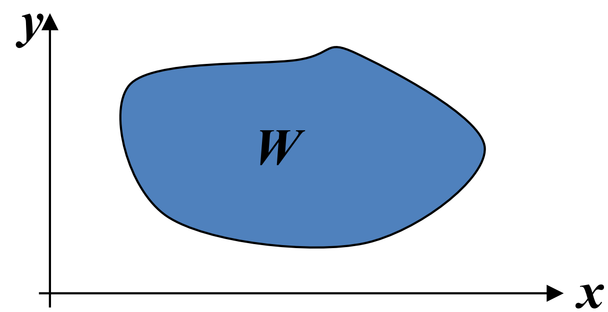
- Additive uncertainty

$$x_{t+1} = Ax_t + Bu_t + w_t$$

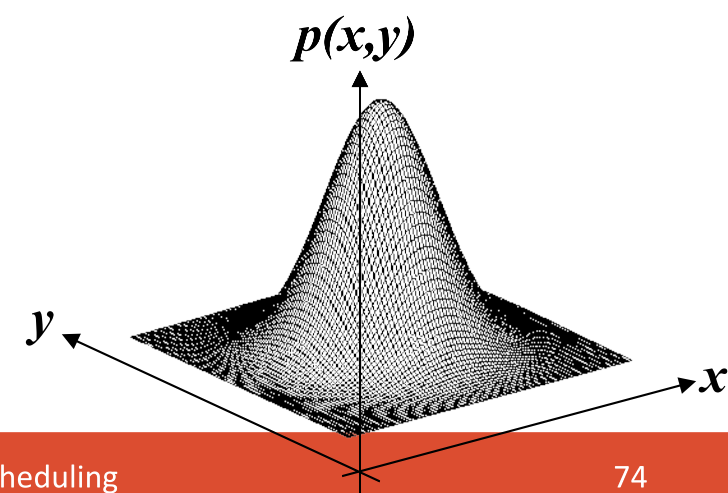
- Multiplicative uncertainty

$$x_{t+1} = (A + \Delta A)x_t + Bu_t$$

$$w_t \in W$$



$$p(w_t) = N(\hat{w}_t, \mathbf{P}_0)$$



Robust Model Predictive Control

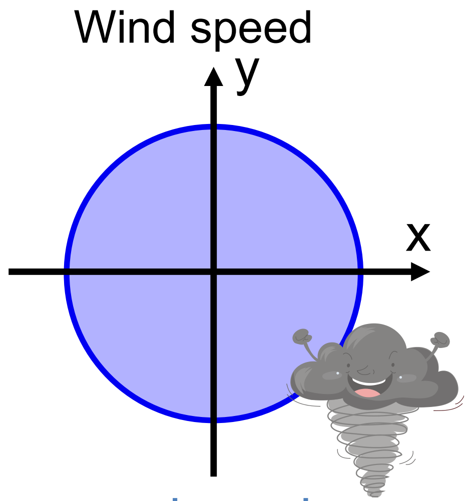
- Receding horizon (MPC) planners react to uncertainty ***after*** something goes wrong.

- Can we take precautionary actions ***before*** something goes wrong?

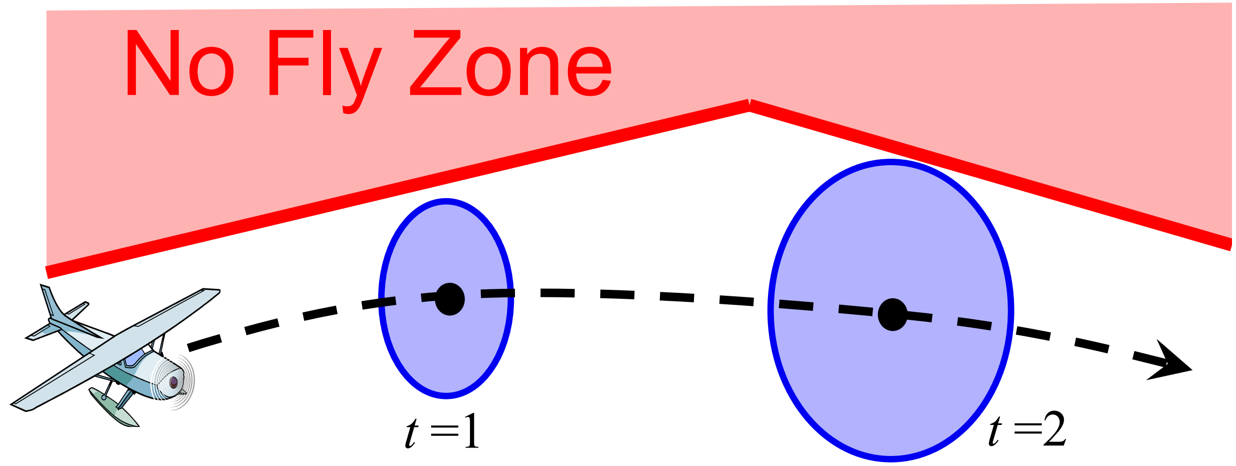
•Ali A. Jalali and Vahid Nadimi, "A Survey on Robust Model Predictive Control from 1999-2006."

Robust versus Chance Constrained

Robust Predictive Control

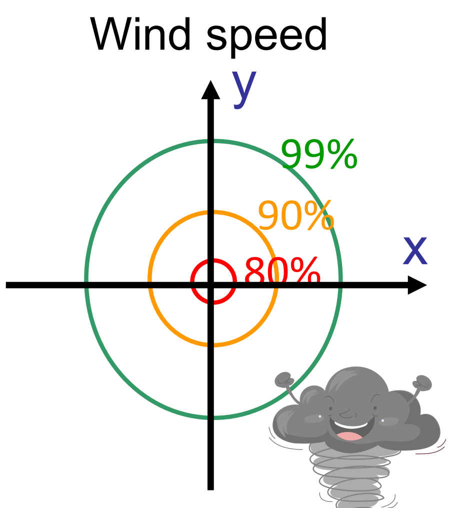


Assume bounds on uncertainty.

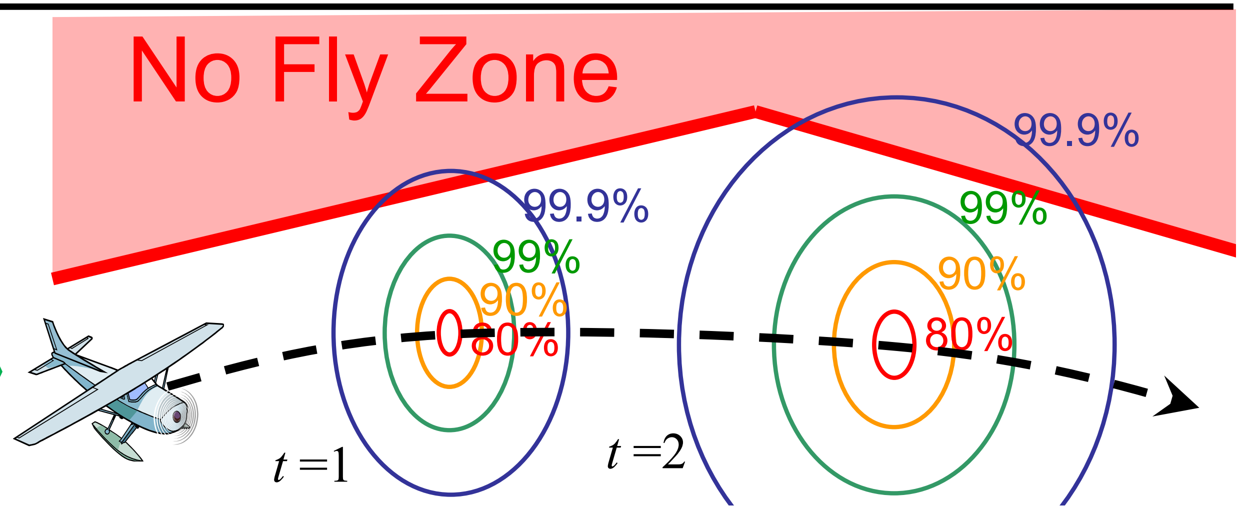


- Predicted position has bounded uncertainty.
- Problem: Find control sequence that satisfies constraints for all realizations of uncertainty.

Chance-constrained Predictive Control



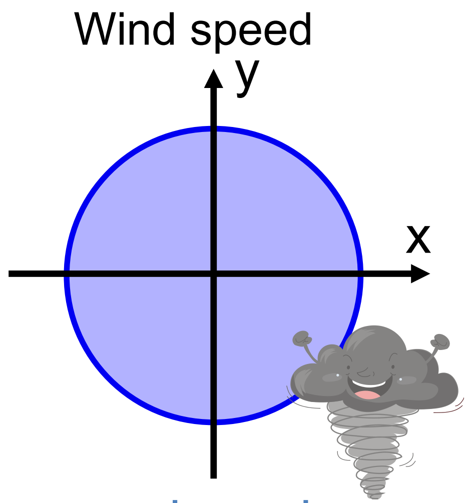
Assume probability distribution that characterizes uncertainty.



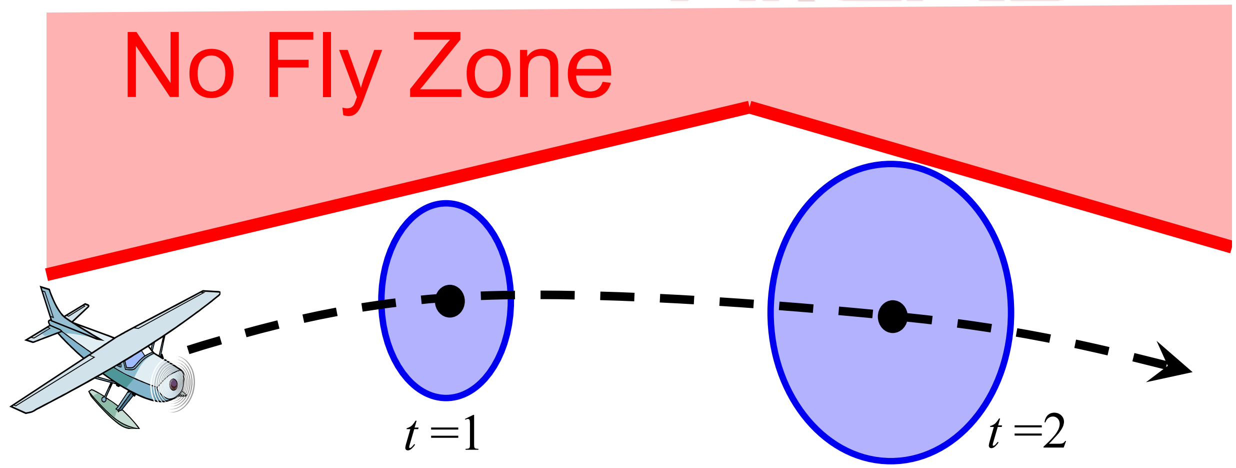
- Predicted position has probabilistic uncertainty.
- Problem: Find control sequence that satisfies constraints within a probability bound (*Chance Constraint*).

Robust versus Chance Constrained

Robust Predictive Control



Assume bounds on uncertainty.



- Predicted position has bounded uncertainty.
- Problem: Find control sequence that satisfies constraints for all realizations of uncertainty.

Chance-constrained Predictive Control

What to Minimize? (*Bounded* Uncertainty)

- Minimize the worst case cost

$$\min_{\mathbf{U}} \max_{w \in W} J(\mathbf{X}, \mathbf{U})$$

$$s.t. \quad \forall_{w \in W} \; h_t^{iT} x_t \leq g_t^i$$

$$w \in W : \text{Bounded uncertainty}$$

- Minimize nominal cost

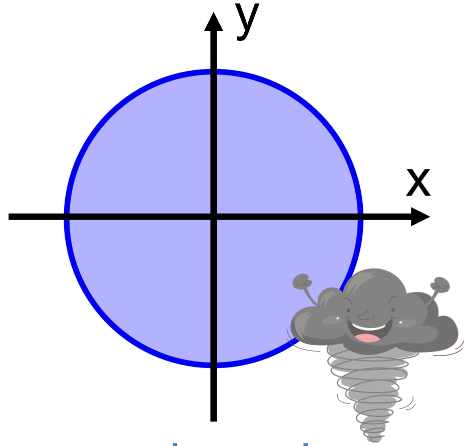
$$\min_{\mathbf{U}} J(\bar{\mathbf{X}}, \mathbf{U}) : \text{Cost when } w = \mathbf{0}$$

$$s.t. \quad \forall_{w \in W} \; h_t^{iT} x_t \leq g_t^i$$

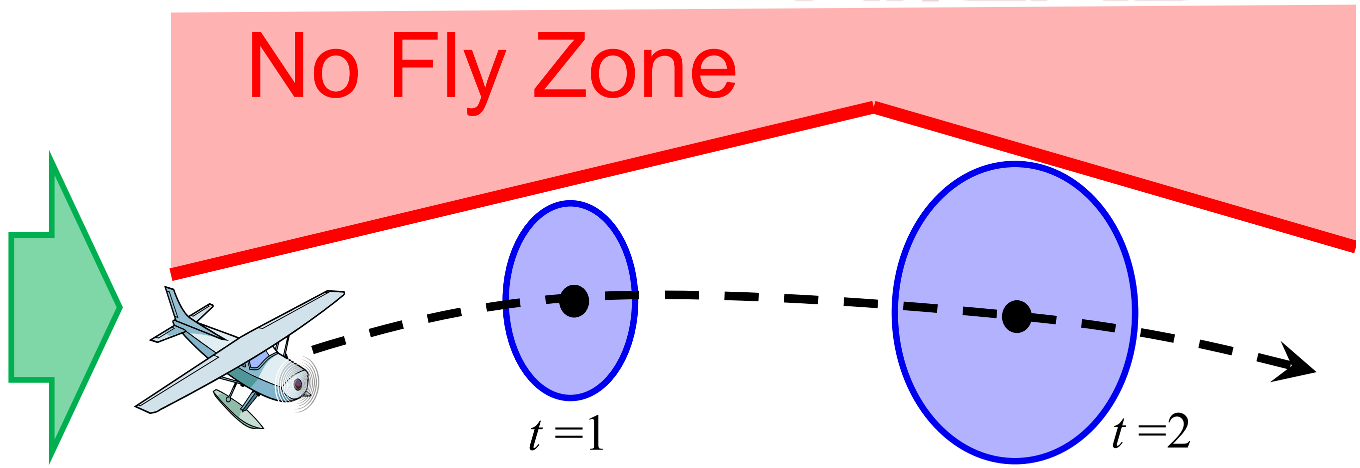
$$w \in W : \text{Bounded uncertainty}$$

Robust versus Chance Constrained

Robust Predictive Control

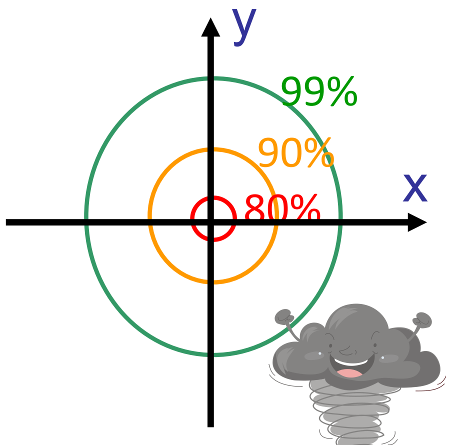


Assume bounds on uncertainty.

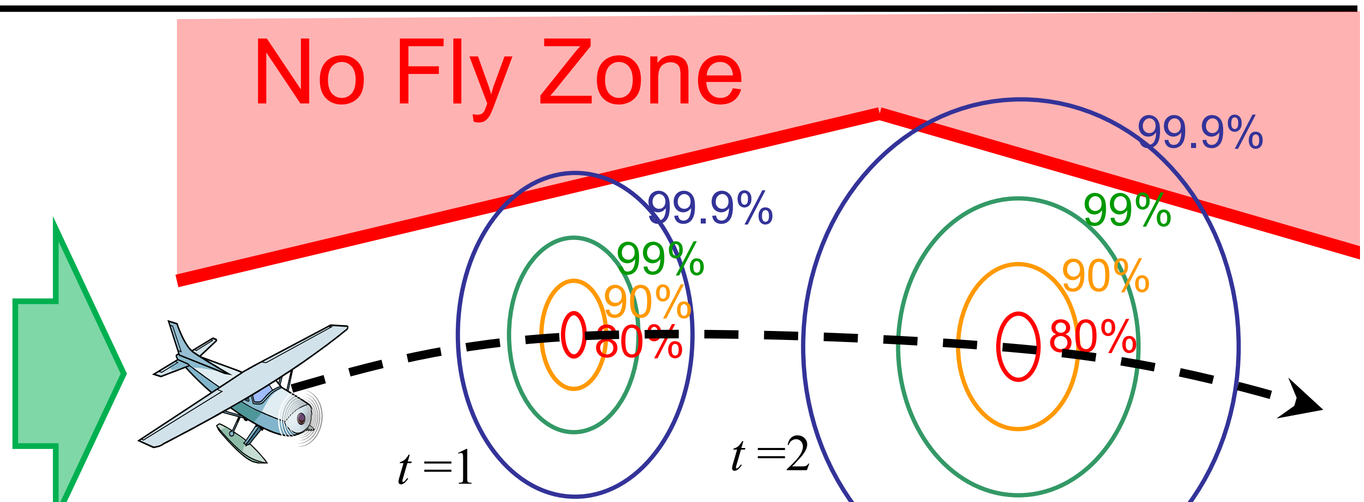


- Predicted position has bounded uncertainty.
- Problem: Find control sequence that satisfies constraints for all realizations of uncertainty.

Chance-constrained Predictive Control



Assume probability distribution that characterizes uncertainty.

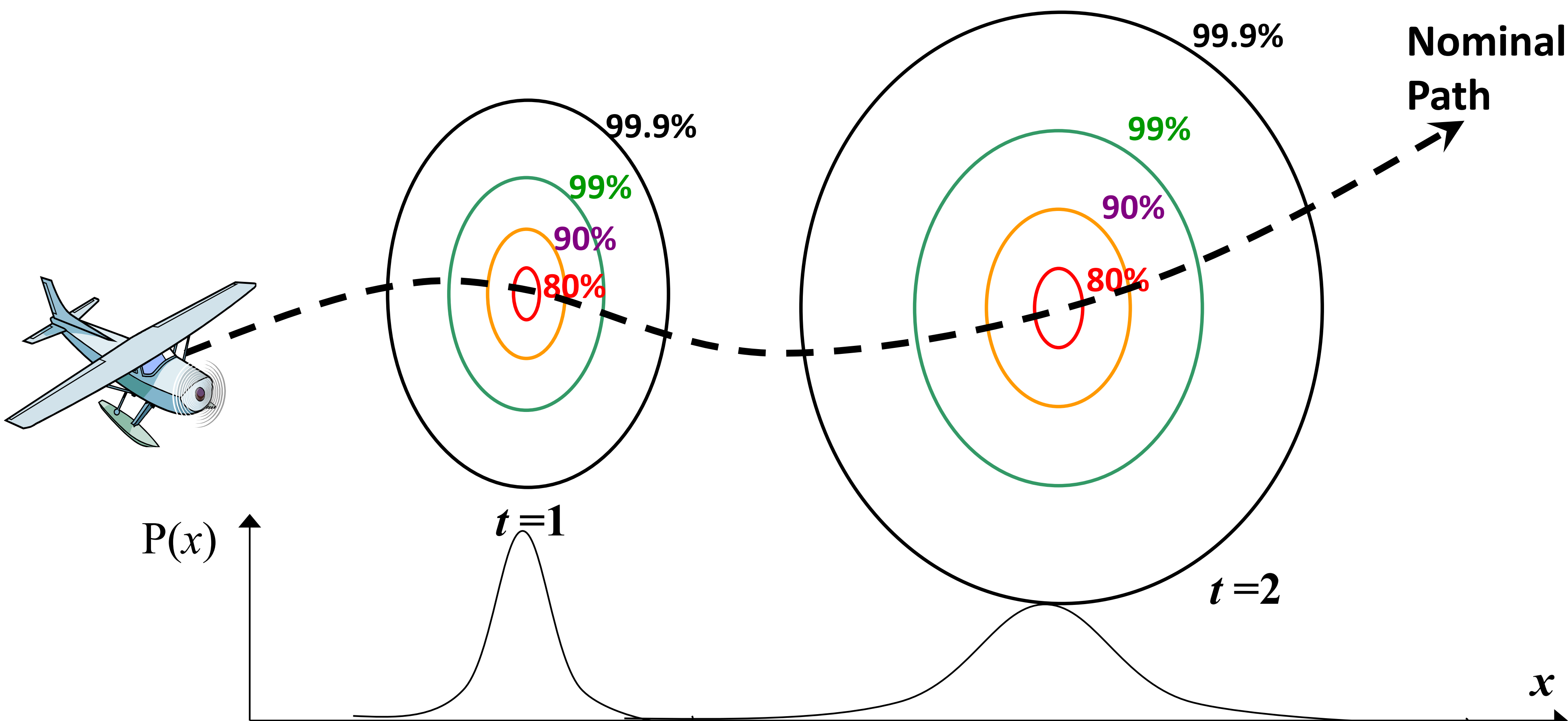


- Predicted position has probabilistic uncertainty.
- Problem: Find control sequence that satisfies constraints within a probability bound (*Chance Constraint*).

Optimal control Under Stochastic Uncertainty

- Exogenous disturbance
- State estimation error

Risk of constraint violation



What to Minimize? (*Stochastic* Uncertainty)

- Utilitarian approach

$$\min_{\mathbf{U}} J(\bar{\mathbf{X}}, \mathbf{U}) + \underbrace{p}_{\text{Penalty (constant)}}\underbrace{f(\mathbf{U})}_{\text{Probability of failure}}$$

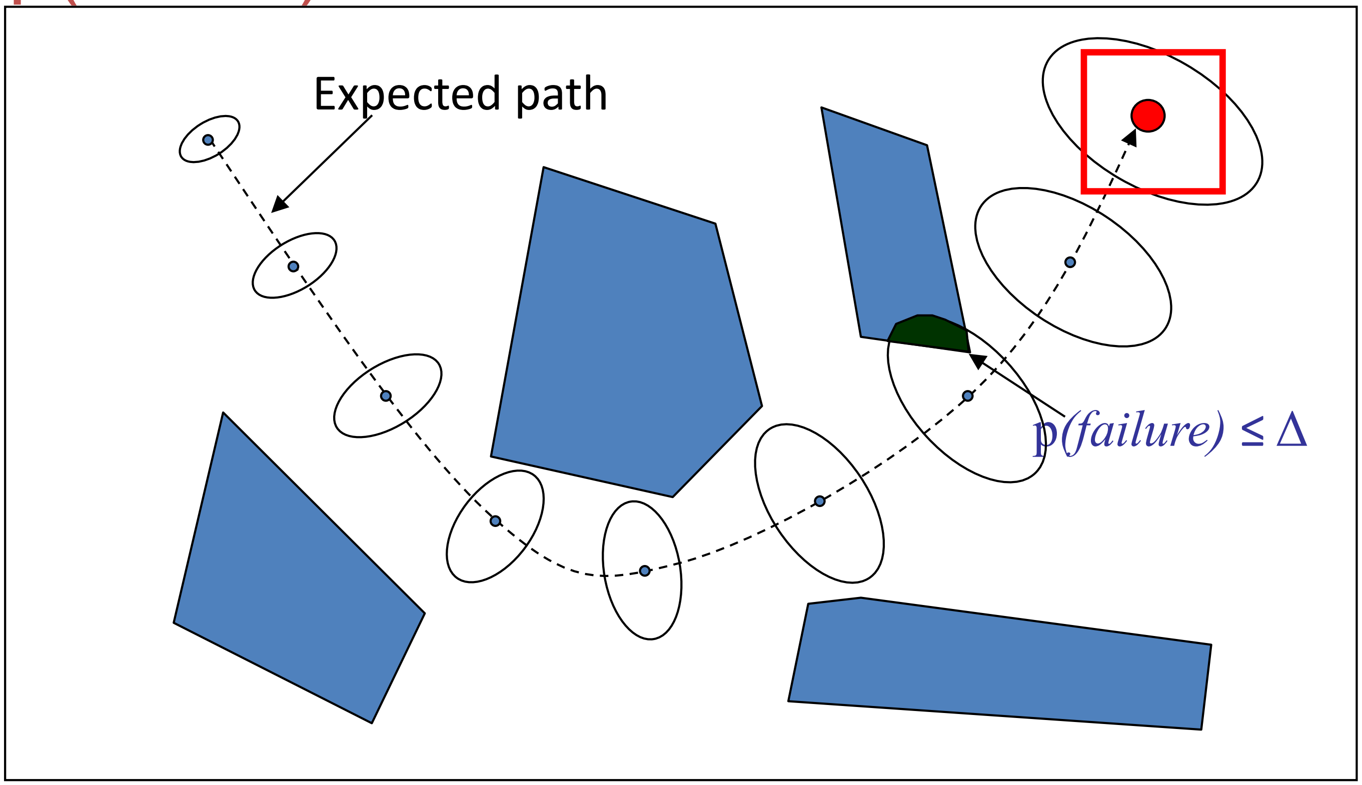
- **Chance constrained optimization**

$$\min_{\mathbf{U}} J(\bar{\mathbf{X}}, \mathbf{U})$$

$$s.t. \quad \underbrace{f(\mathbf{U})}_{\text{Probability of failure}} \leq \underbrace{\Delta}_{\text{Risk bound}}$$

"Risk-bounded" Robust Path Planning

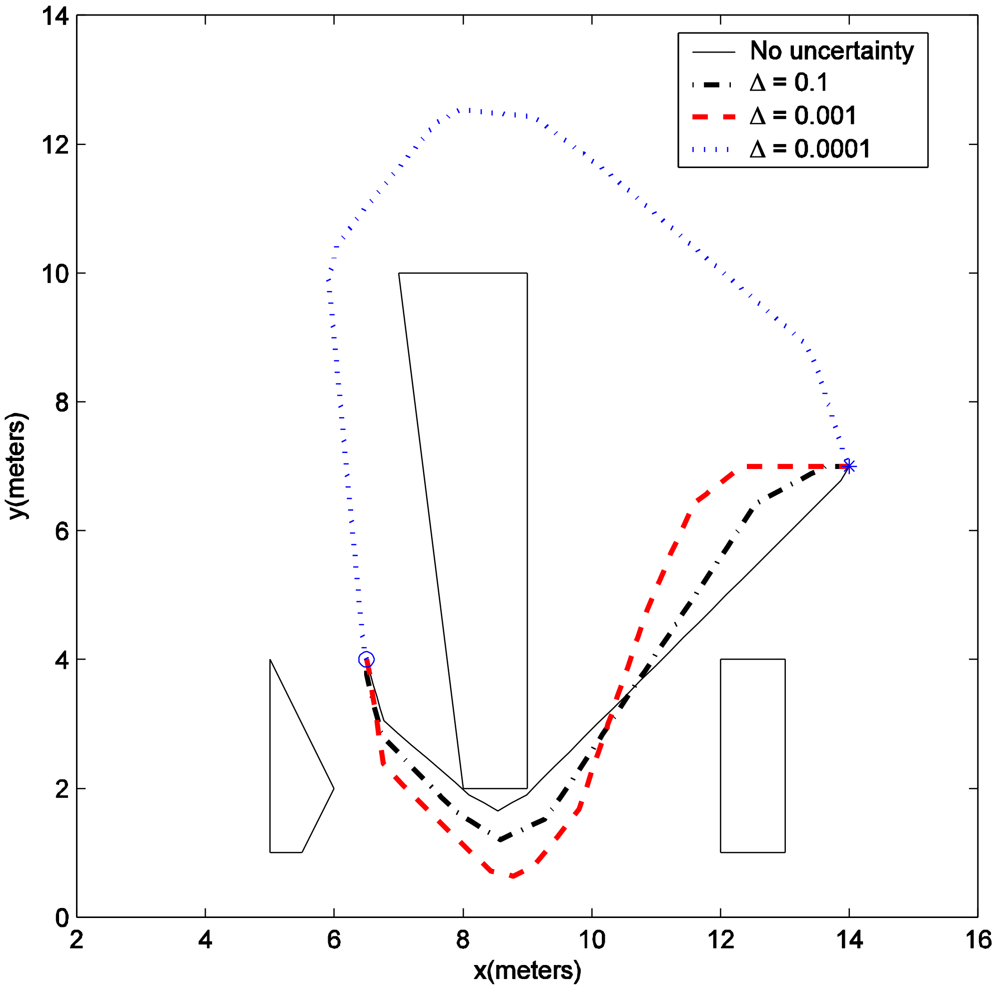
- "Plan optimal path to goal such that p(failure) ≤ Δ."

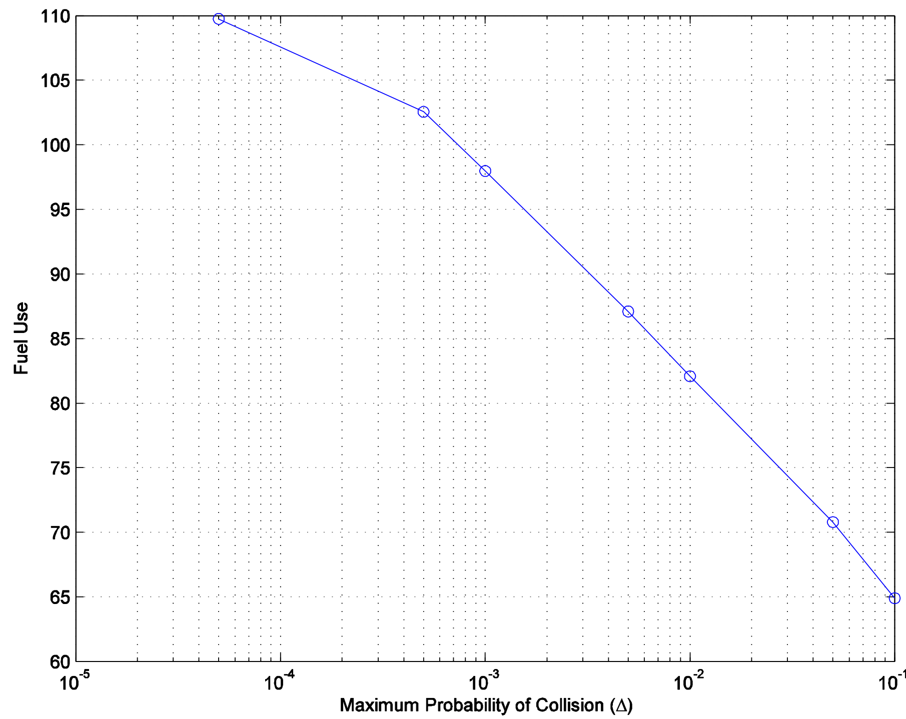


- Called a *Chance Constraint(CC)*

Risk – Performance Tradeoff

- Maximum probability of failure is used to trade performance against risk-aversion.





Method: Uniform Risk Allocation [Blackmore, PhD]

Deterministic Finite-Horizon Optimal Control

$$\min_{u_{1:T} \in \mathbf{U}^T} \underline{J(u_{1:T})}$$

Convex function

Cost function (e.g. fuel consumption)

$$s.t.$$

Discrete-time linear dynamics

$$\bigwedge_{t=0}^{T-1} x_{t+1} = Ax_t + Bu_t$$

State constraints (*Convex*)

$$\bigwedge_{t=1}^{T} \bigwedge_{i=1}^{N} h_t^{iT} x_t \le g_t^i$$

Notations:

$$\bigwedge_{i=1}^{N} C_i \equiv C_1 \wedge C_2 \wedge \ldots \wedge C_N$$

Logical conjunctions

$$\bigvee_{i=1}^{N} C_i \equiv C_1 \vee C_2 \vee \ldots \vee C_N$$

Logical disjunctions

Chance-Constrained FH Optimal Control

$$\min_{u_{1:T} \in \mathbf{U}^T} J(u_{1:T})$$

$$s.t.$$

Stochastic dynamics

$$\bigwedge_{t=0}^{T-1} x_{t+1} = Ax_t + Bu_t + w_t$$

Gaussian distribution

$$w_t \sim N(0, \Sigma_t)$$

Exogenous disturbance

State constraints

$$\bigwedge_{t=1}^{T} \bigwedge_{i=1}^{N} h_t^{iT} x_t \le g_t^i$$

$$x_0 \sim N(\bar{x}_0, \Sigma_{x,0})$$

State estimation error

Chance-Constrained FH Optimal Control

$$\min_{u_{1:T} \in \mathbf{U}^T} J(u_{1:T})$$

$$s.t.$$

Stochastic dynamics
$$\bigwedge_{t=0}^{T-1} x_{t+1} = Ax_t + Bu_t + w_t$$

$$w_t \sim N(0, \Sigma_t)$$

$$x_0 \sim N(\bar{x}_0, \Sigma_{x,0})$$

State constraints
$$\bigwedge_{t=1}^{T} \bigwedge_{i=1}^{N} h_t^{iT} x_t \le g_t^i$$

Chance-Constrained FH Optimal Control

$$\min_{u_{1:T} \in \mathbf{U}^T} J(u_{1:T})$$

$$s.t.$$

Stochastic dynamics

$$\bigwedge_{t=0}^{T-1} x_{t+1} = Ax_t + Bu_t + w_t$$

$$w_t \sim N(0, \Sigma_t)$$

$$x_0 \sim N(\bar{x}_0, \Sigma_{x,0})$$

Chance constraint

$$\Pr\left[\bigwedge_{t=1}^{T} \bigwedge_{i=1}^{N} h_t^{iT} x_t \leq g_t^i \right] \geq 1 - \Delta$$

Risk bound
(Upper bound on the probability of failure)
Assumption: $\Delta < 0.5$

Solution Methods for Chance-Constrained Problems

- Sampling based methods
  - Scenario-based
    - Bernardini and Bemporad, 2009
  - Particle control
    - Blackmore et al., 2010
- Non-sampling-based methods
  - Elliptic approximation
    (direct extension of robust predictive control)
    - van Hessem, 2004
  - Risk allocation
    - Ono and Williams, 2008

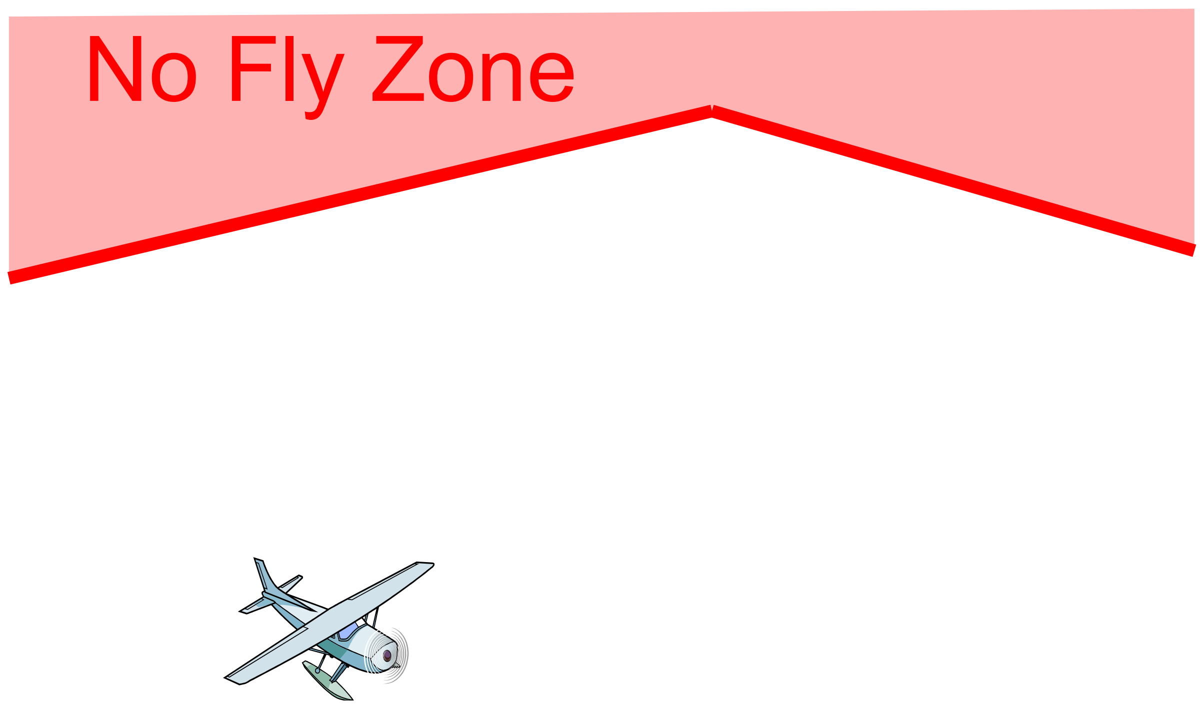
Outline

- A Risky Business
- Overview of Risk-bounded Planning
- Goal-directed Trajectory Planning
- Risk-bounded Trajectory Planning
  - Problem Statement
  - Elliptic Approximation
  - Iterative Risk Allocation (IRA)
  - Convex Risk Allocation (CRA)

Elliptic Approximation

Chance constraint:

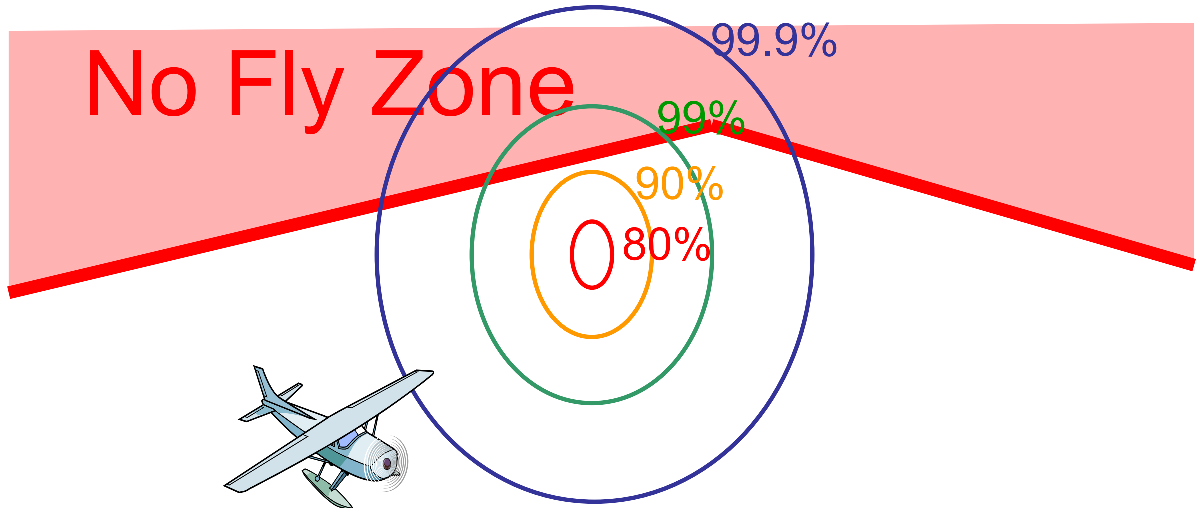
Risk < 1%



Elliptic Approximation

Chance constraint:

Risk < 1%



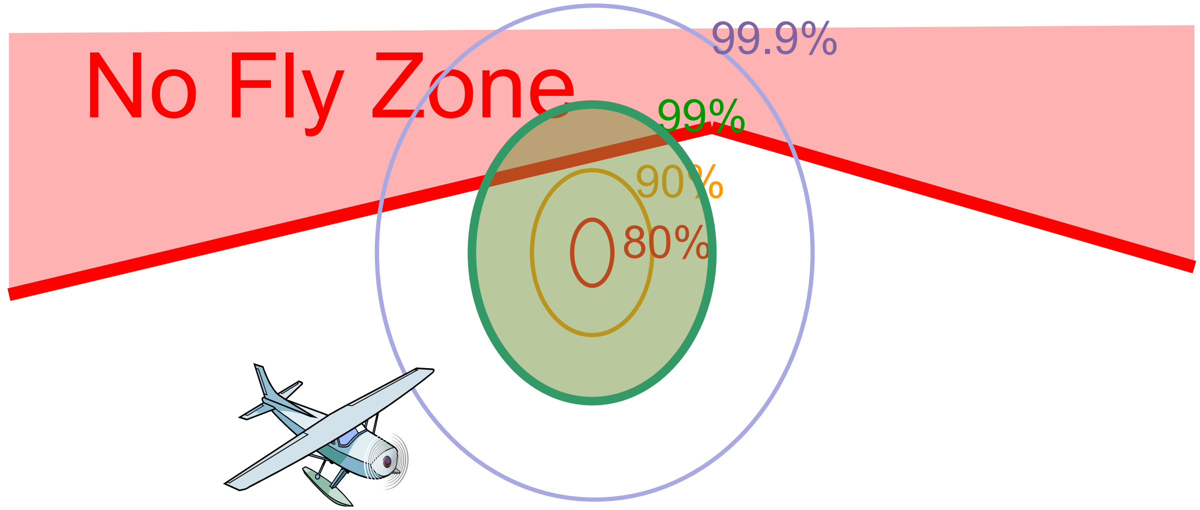
1. Derive probability distribution over future states as a function of control inputs.

Note: When planning in an N-dimensional state space over time steps,
a joint distribution over an N-dimensional space must be considered.

Elliptic Approximation

Chance constraint:

Risk < 1%

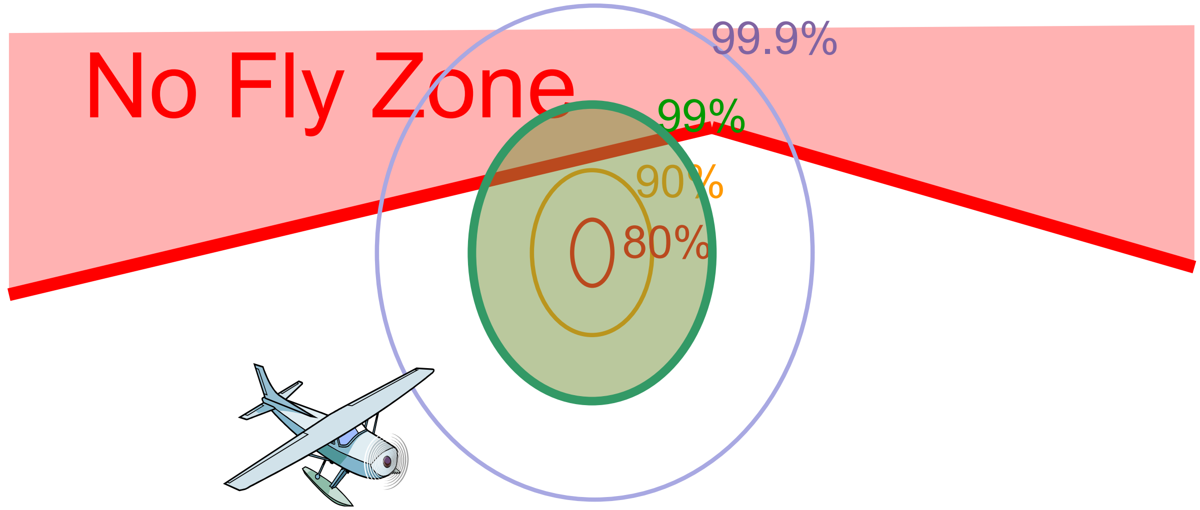


1. Derive probability distribution over future states as a function of control inputs.
2. Find a 99% probability ellipse.

Elliptic Approximation

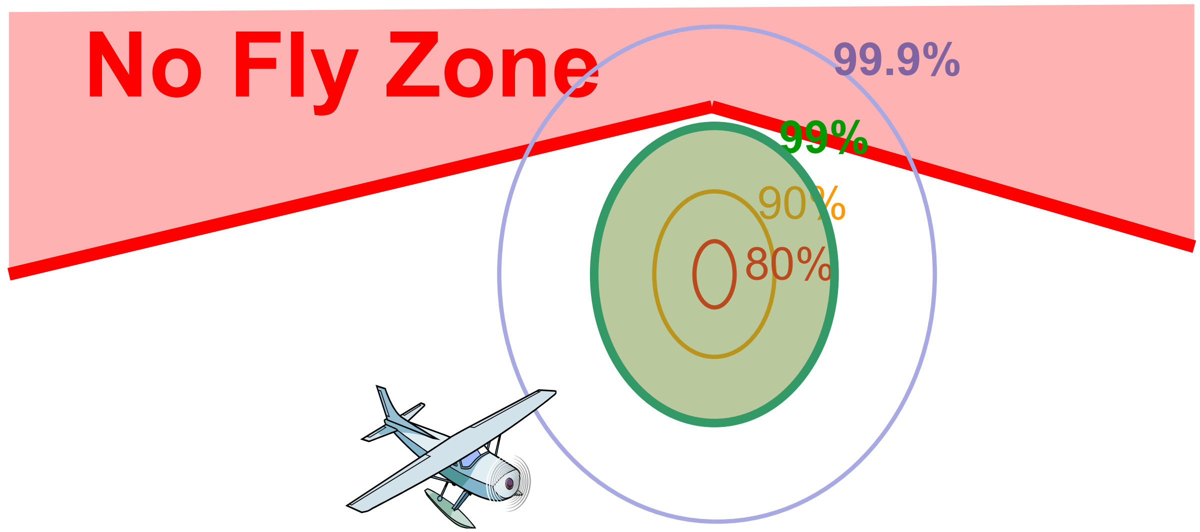
Chance constraint:

Risk < 1%



1. Derive probability distribution over future states as a function of control inputs.
2. Find a 99% probability ellipse.
3. Find control sequence such that the probability ellipse is within the constraint boundaries.

Conservatism of Elliptic Approximation



Issue: often *very* conservative

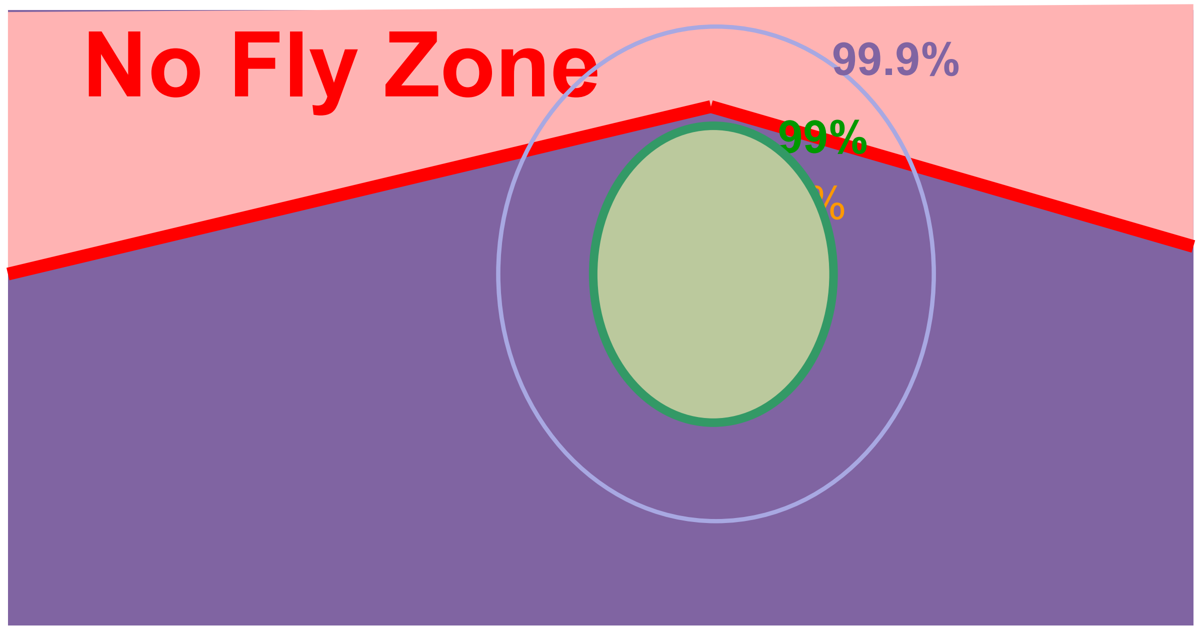
Real probability of failure $= \int p(x)dx$

Probability density function

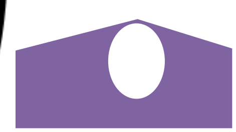
$$< 1 - \int p(x)dx = 1\%$$

Conservatism of Elliptic Approximation

Issue: often *very* conservative.



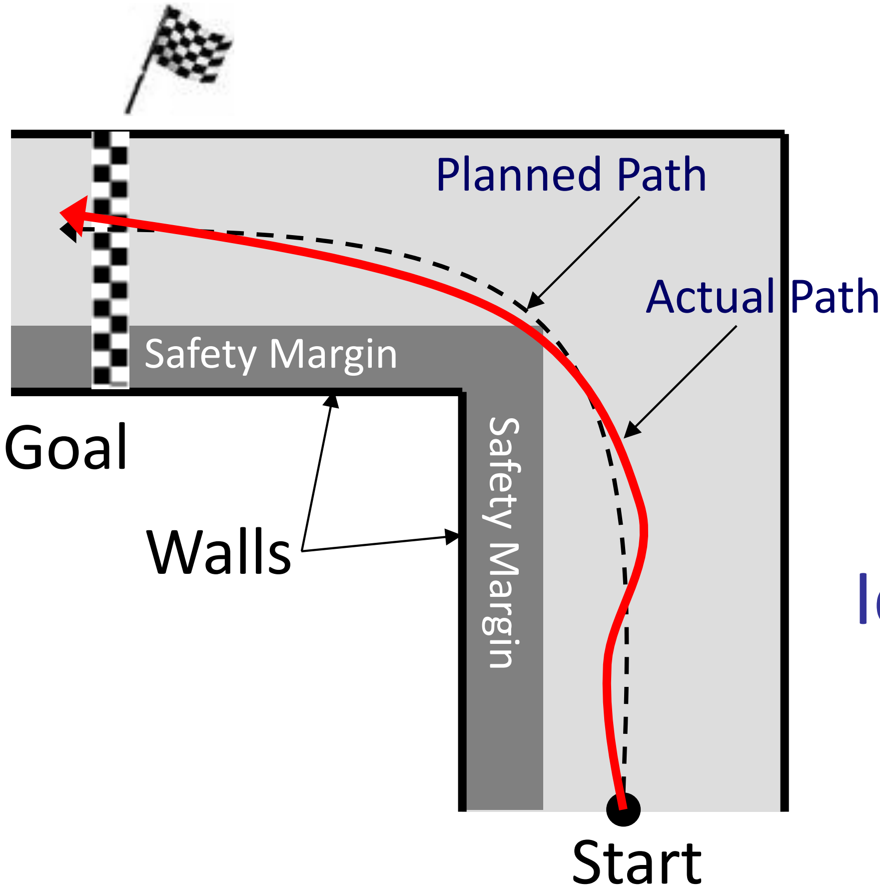
$$\text{Conservatism} = \int p(x)\,dx$$



Outline

- A Risky Business
- Overview of Risk-bounded Planning
- Goal-directed Trajectory Planning
- Risk-bounded Trajectory Planning
  - Problem Statement
  - Elliptic Approximation
  - Iterative Risk Allocation (IRA)
  - Convex Risk Allocation (CRA)

Example: Race Car Path Planning



<u>Problem</u>

Find the fastest path to the goal, while limiting the probability of crash throughout the race *to 0.1%*

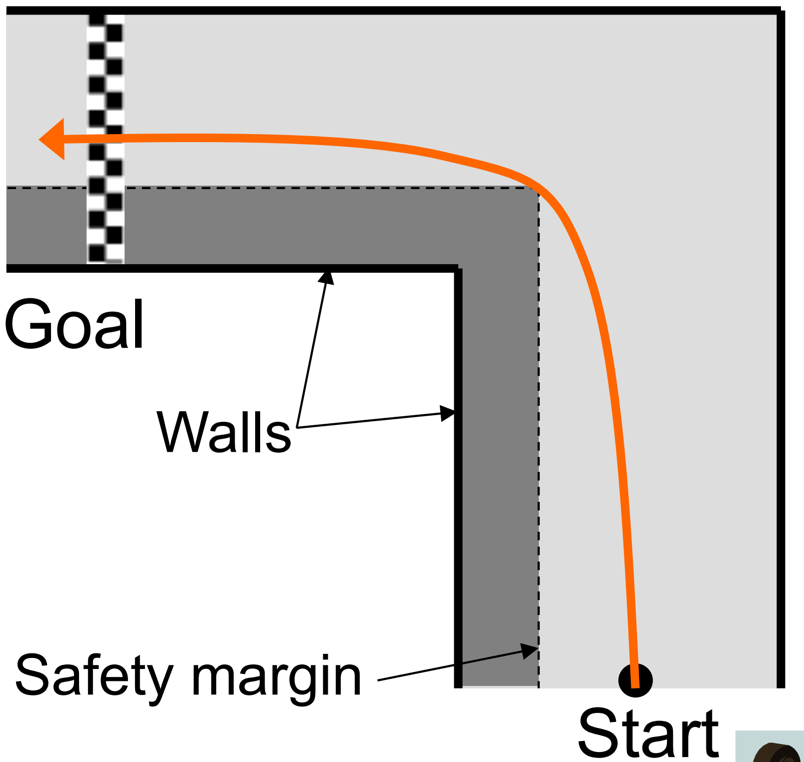
Risk bound

Idea:

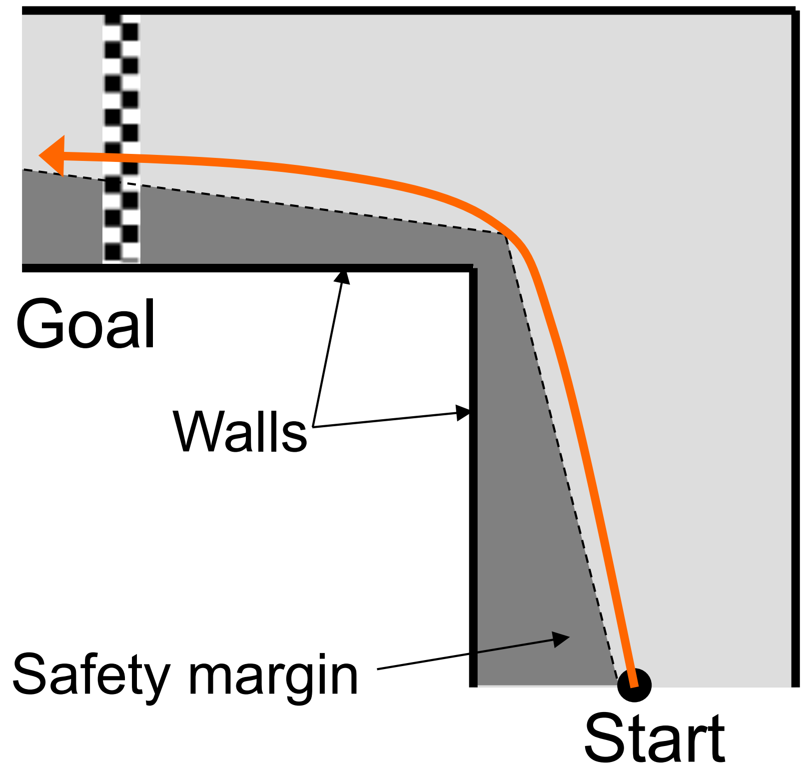
1. Create safety margin that satisfies the risk bound from start to the goal.
2. Reduce to simpler, deterministic optimization problem.

Idea: Generate safety margin that satisfies risk bound while maximizing expected utility

(a) Uniform width safety margin



(b) Uneven width safety margin



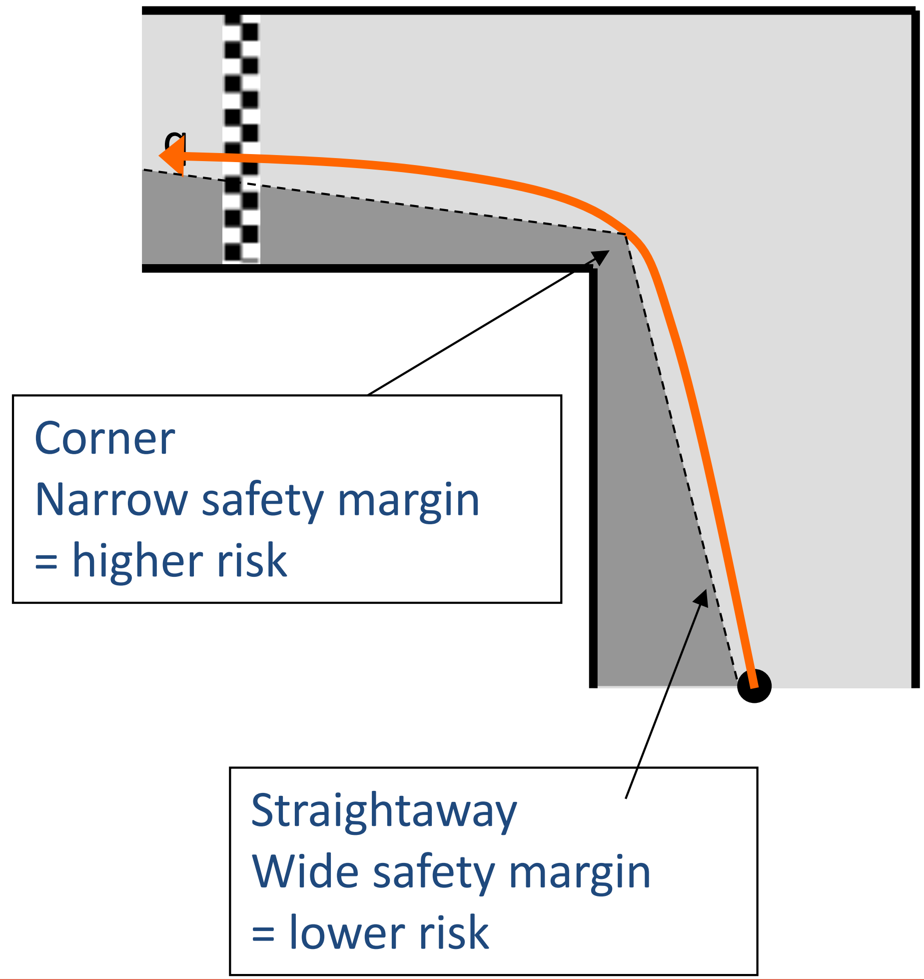
(b) results in better path → takes risk when most beneficial

Approach: Algorithmic Risk Allocation

[Ono & Williams, AAAI 08]

Key Idea - *Risk Allocation*

- Taking more risk at the corner results in a shorter path, than taking the same risk at the straightaway.
- **Sensitivity** of path length to risk **is higher at the corner.**
- ***Risk Allocation***
  - Optimize allocation of risk to time steps and constraints.

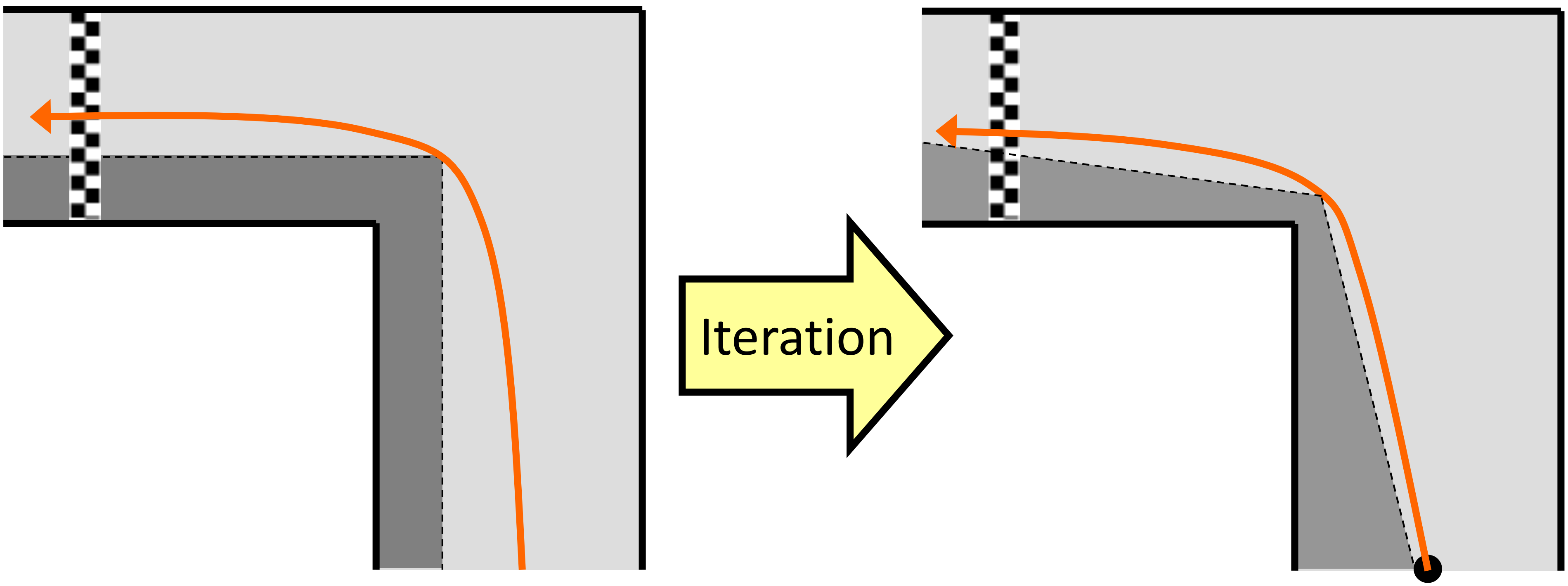


Iterative Risk Allocation (IRA) Algorithm

- Descent algorithm

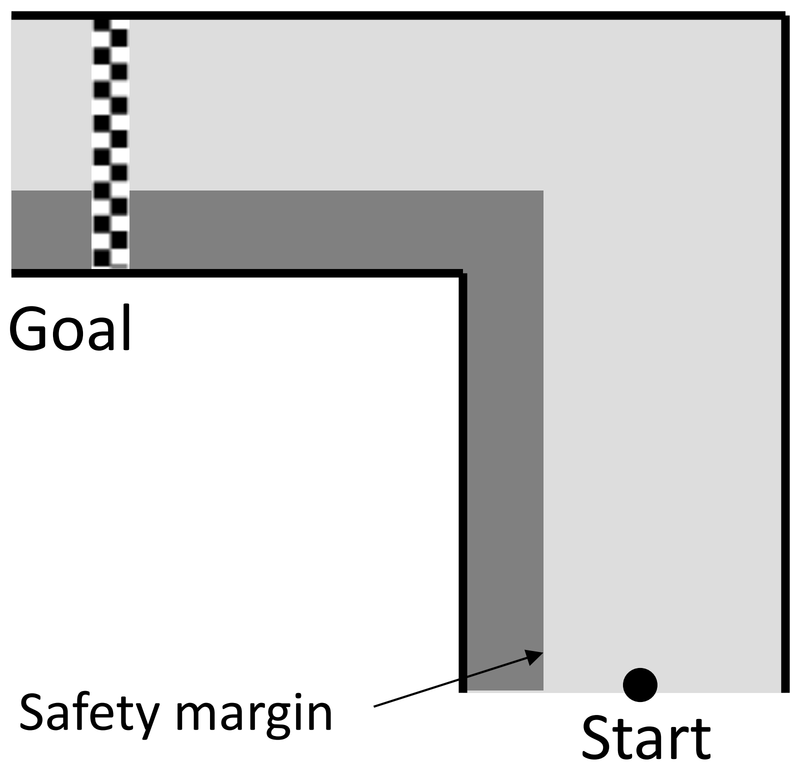
$$\bar{J}^*(\boldsymbol{\delta}_0) \geq \bar{J}^*(\boldsymbol{\delta}_1) \geq \bar{J}^*(\boldsymbol{\delta}_2) \cdots$$

(Refer to paper for proof)



- Starts from a suboptimal risk allocation
- Improves it by iteration

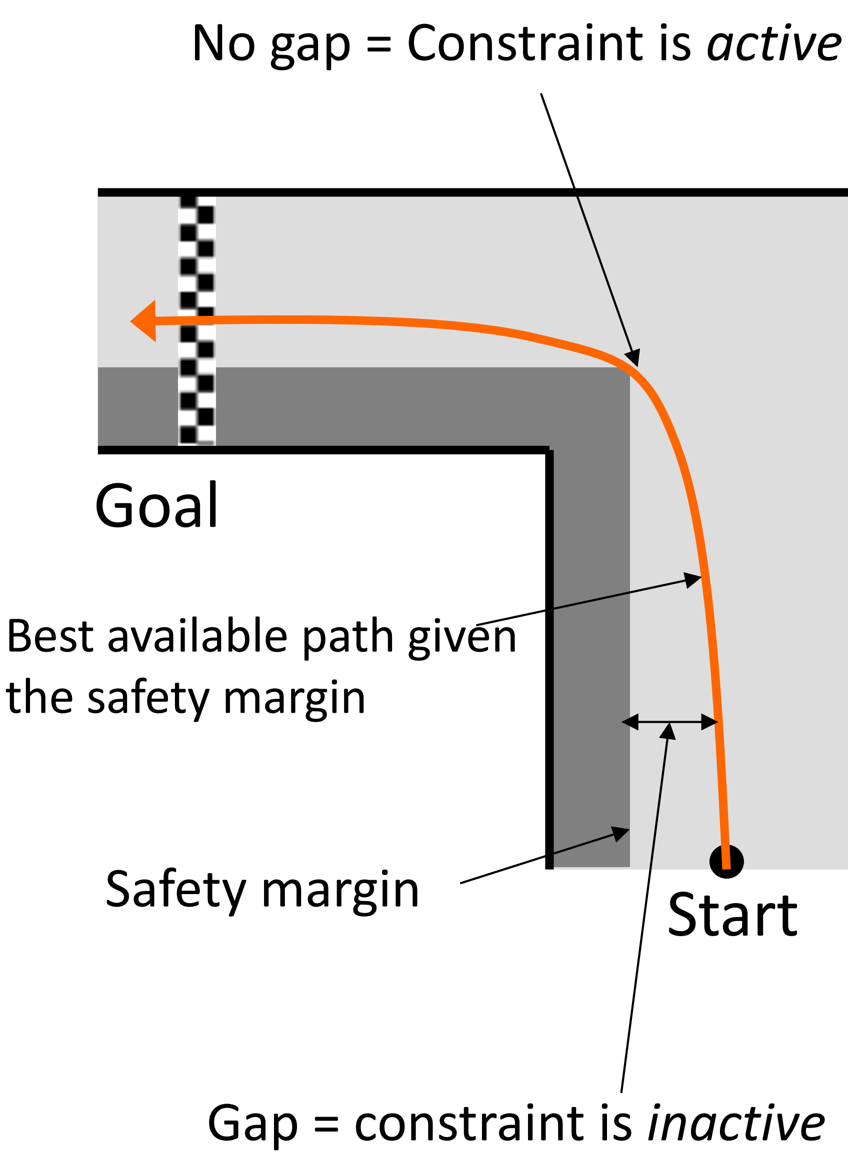
Iterative Risk Allocation Algorithm



Algorithm IRA

1 Initialize with arbitrary risk allocation
2 Loop
3 Compute the best available path given the current risk allocation
4 Decrease the risk where the constraint is inactive
5 Increase the risk where the constraint is active
6 End loop

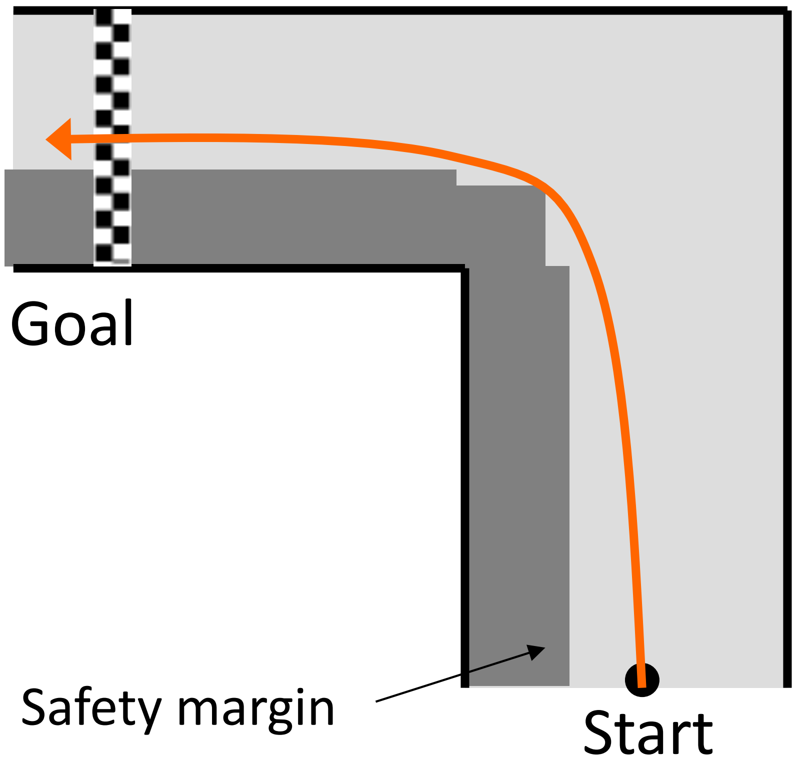
Iterative Risk Allocation Algorithm



Algorithm IRA

1 Initialize with arbitrary risk allocation
2 Loop
3 Compute the best available path given the current risk allocation
4 Decrease the risk where the constraint is inactive
5 Increase the risk where the constraint is active
6 End loop

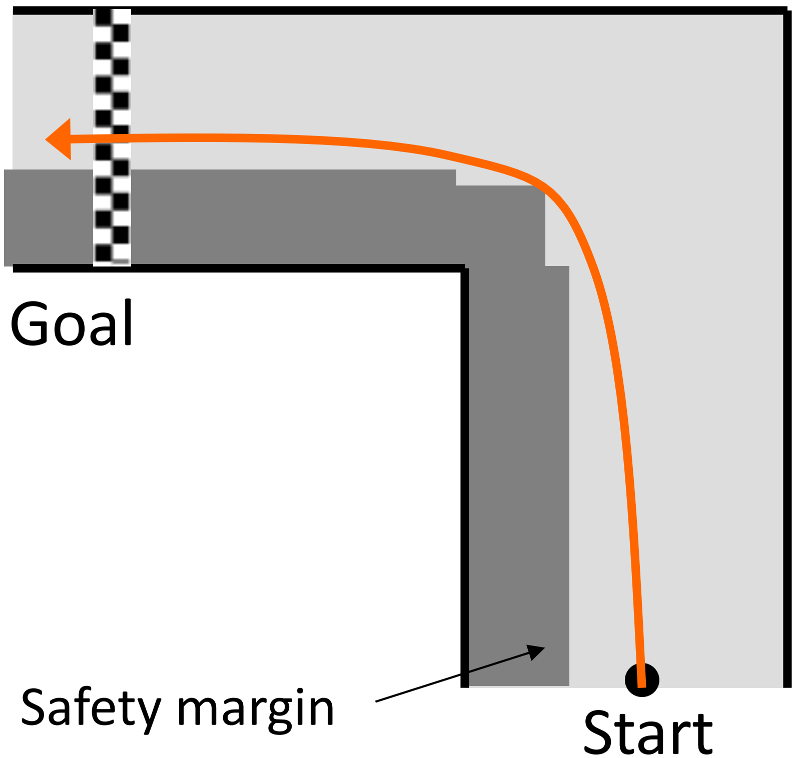
Iterative Risk Allocation Algorithm



Algorithm IRA

1 Initialize with arbitrary risk allocation
2 Loop
3 Compute the best available path given the current risk allocation
4 Decrease the risk where the constraint is inactive
5 Increase the risk where the constraint is active
6 End loop

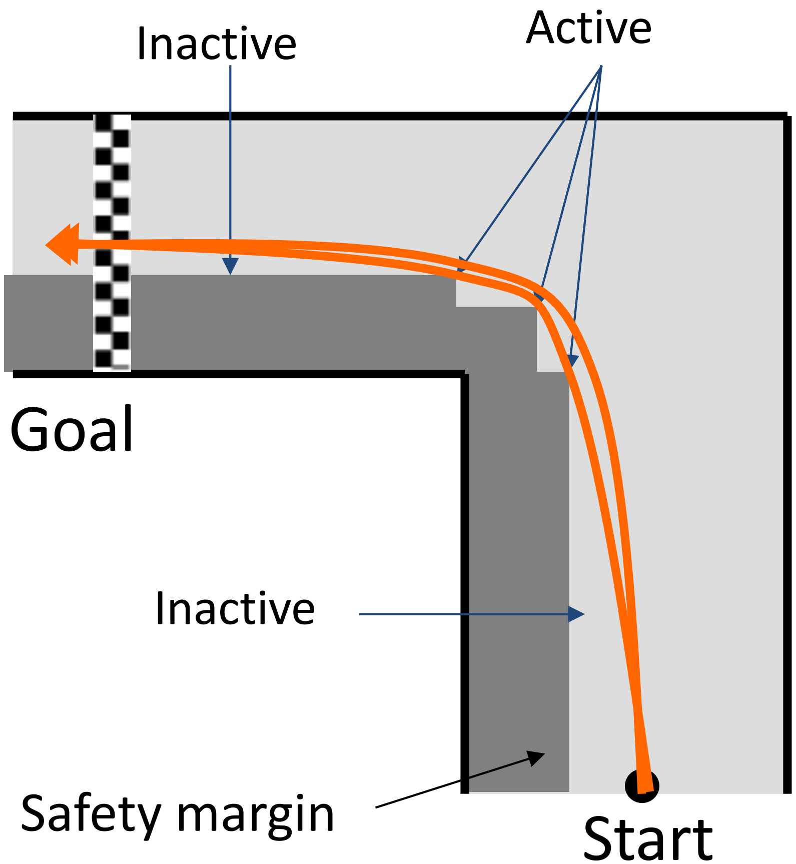
Iterative Risk Allocation Algorithm



Algorithm IRA

1 Initialize with arbitrary risk allocation
2 Loop
3 Compute the best available path given the current risk allocation
4 Decrease the risk where the constraint is inactive
<u>**5**</u> Increase the risk where the constraint is active
6 End loop

Iterative Risk Allocation Algorithm



Algorithm IRA

1 Initialize with arbitrary risk allocation

2 Loop

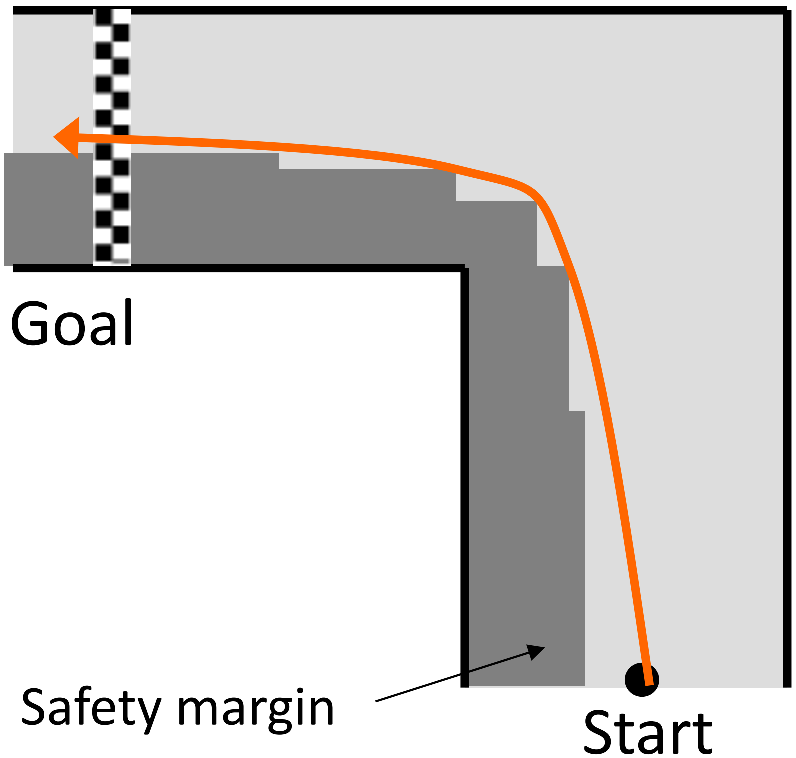
3 Compute the best available path given the current risk allocation

4 Decrease the risk where the constraint is inactive

5 Increase the risk where the constraint is active

6 End loop

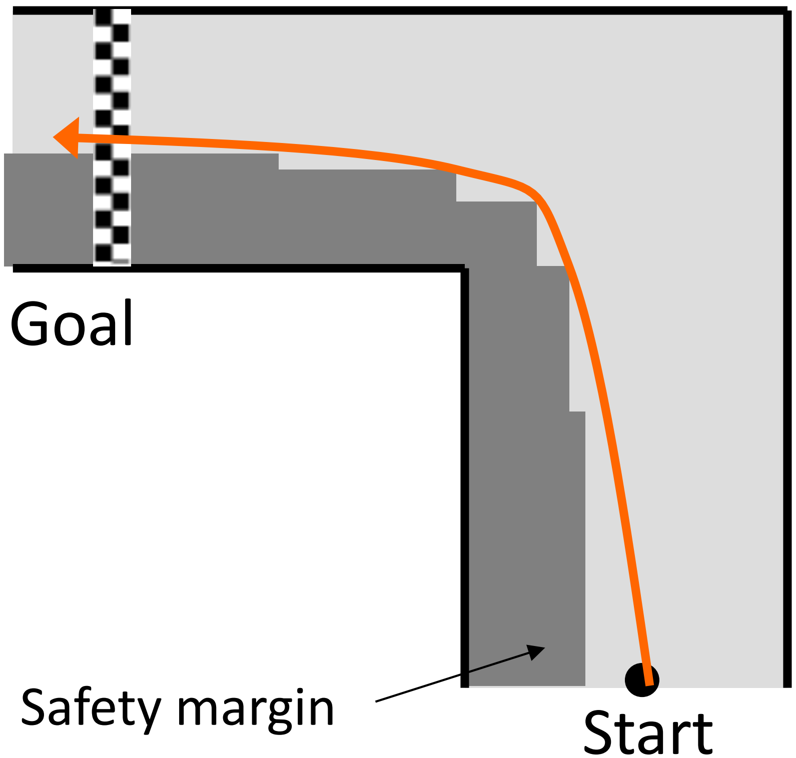
Iterative Risk Allocation Algorithm



Algorithm IRA

1 Initialize with arbitrary risk allocation
2 Loop
3 Compute the best available path given the current risk allocation
4 Decrease the risk where the constraint is inactive
5 Increase the risk where the constraint is active
6 End loop

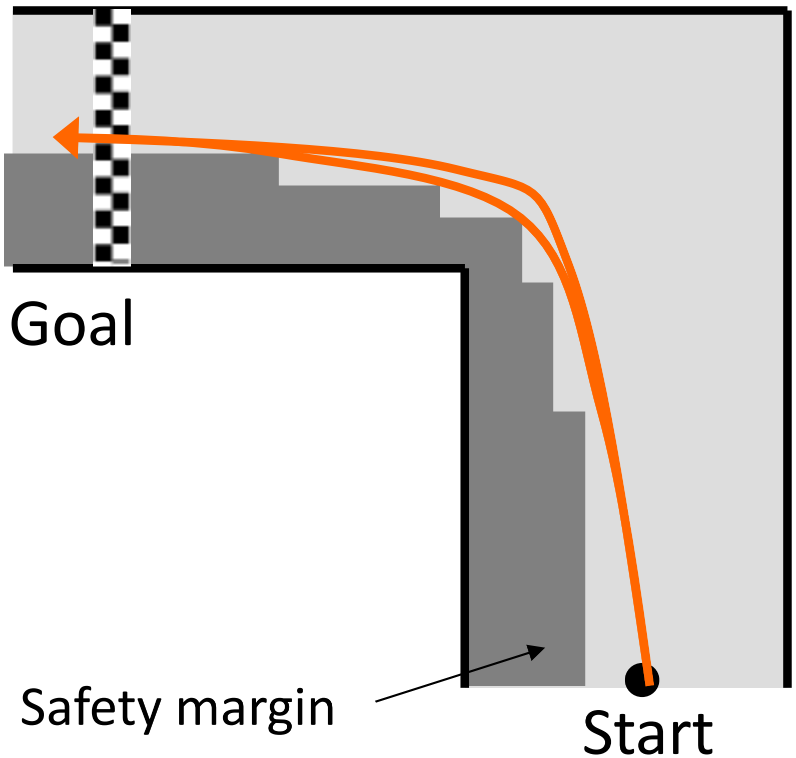
Iterative Risk Allocation Algorithm



Algorithm IRA

1 Initialize with arbitrary risk allocation
2 Loop
3 Compute the best available path given the current risk allocation
4 Decrease the risk where the constraint is inactive
5 Increase the risk where the constraint is active
6 End loop

Iterative Risk Allocation Algorithm



Algorithm IRA

1 Initialize with arbitrary risk allocation

2 Loop

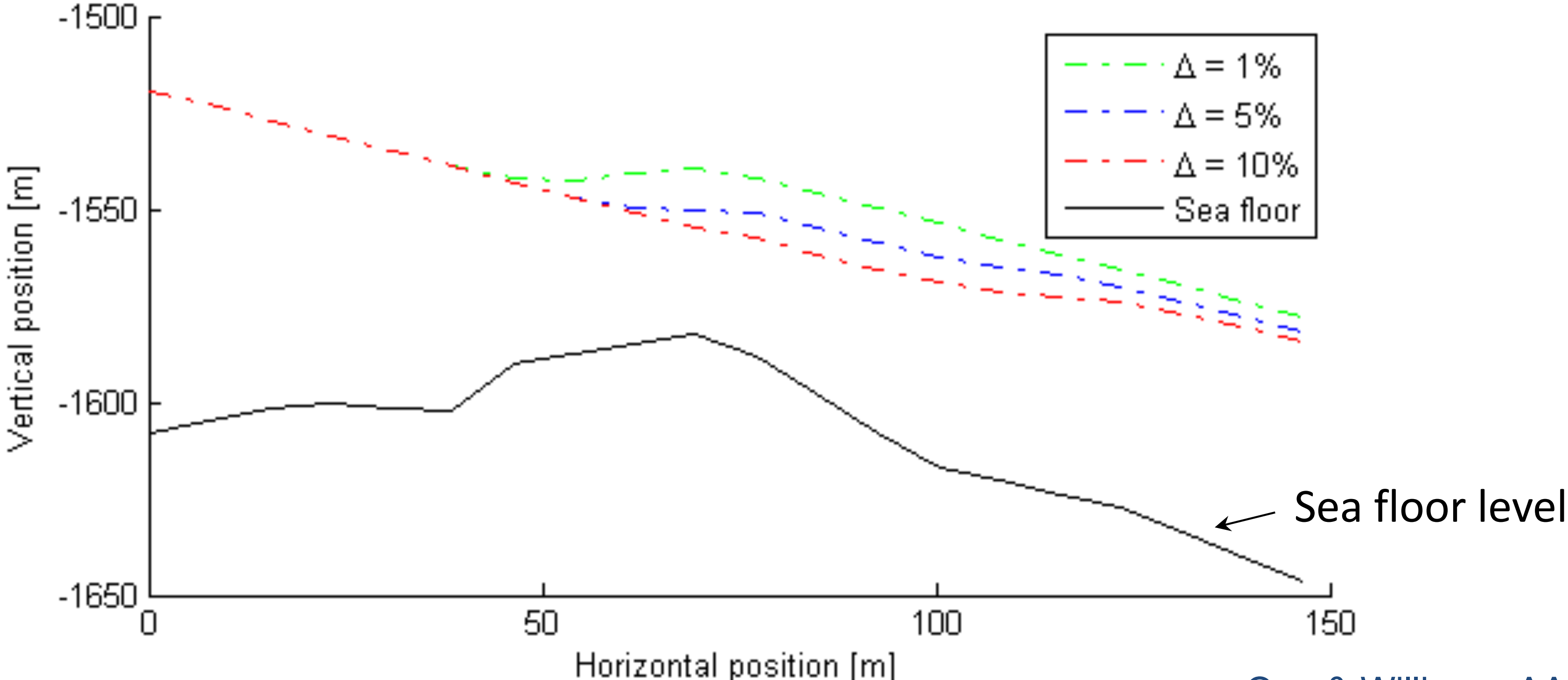
<u>3</u> Compute the best available path given the current risk allocation

4 Decrease the risk where the constraint is inactive

5 Increase the risk where the constraint is active

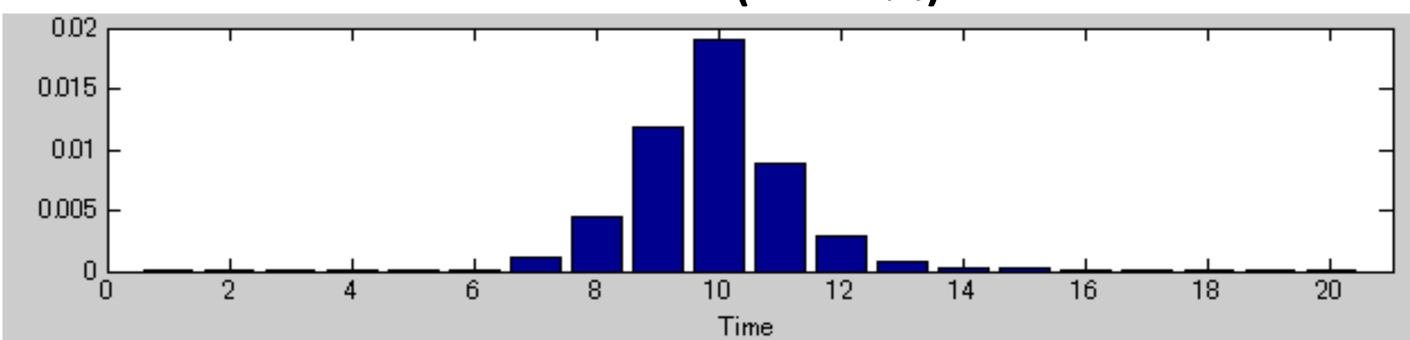
6 End loop

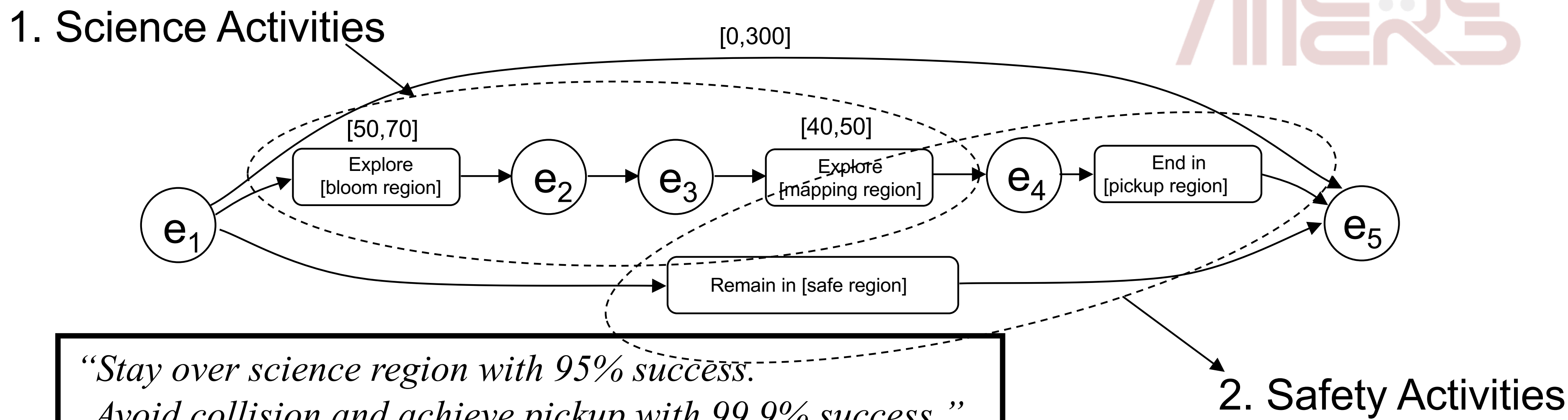
Monterey Bay Mapping Example



Ono & Williams, AAAI 08

Risk allocation ($\Delta = 5\%$)

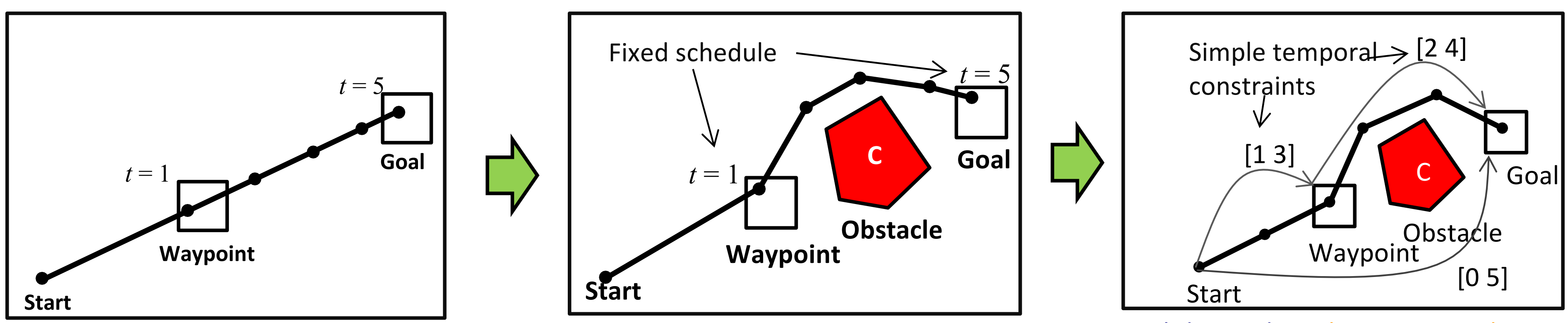




"Stay over science region with 95% success.
Avoid collision and achieve pickup with 99.9% success."

Approach: Solve increasingly expressive problems

(QSP)



Convex chance-constrained traj opt | Non-convex, chance-constrained traj opt | Goal-directed qualitative state plan traj opt

IRA Applies to All Problem Statements!

Approach

- **Determinize problem through risk allocation**

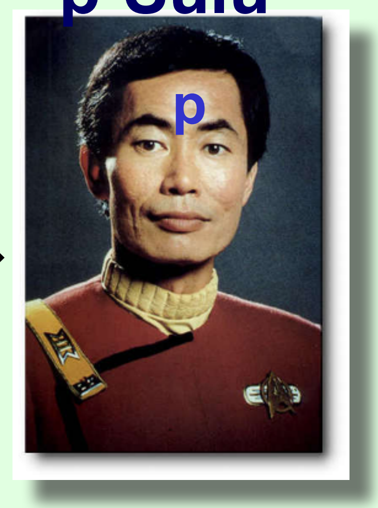
Input

Goal (cc-QSP) +

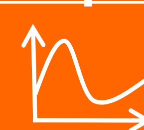
Stochastic Model

($X_{t+1} = AX_t + Bu_t + w_t$)

p-Sulu



Output

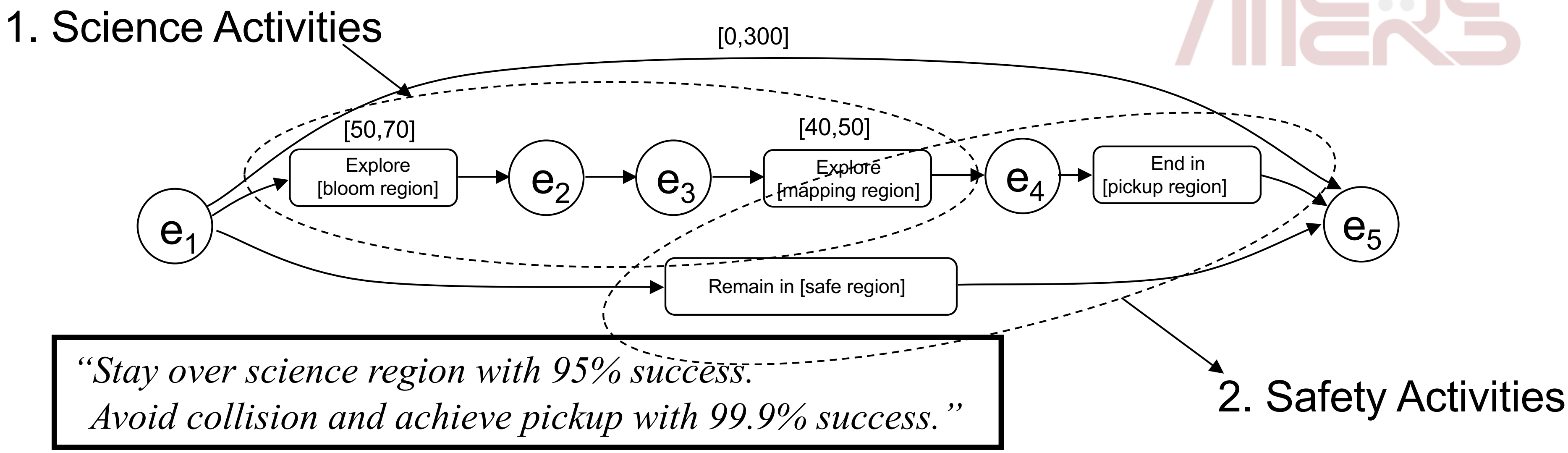


Control Sequence

Constraint or Mathematical Programming Solver

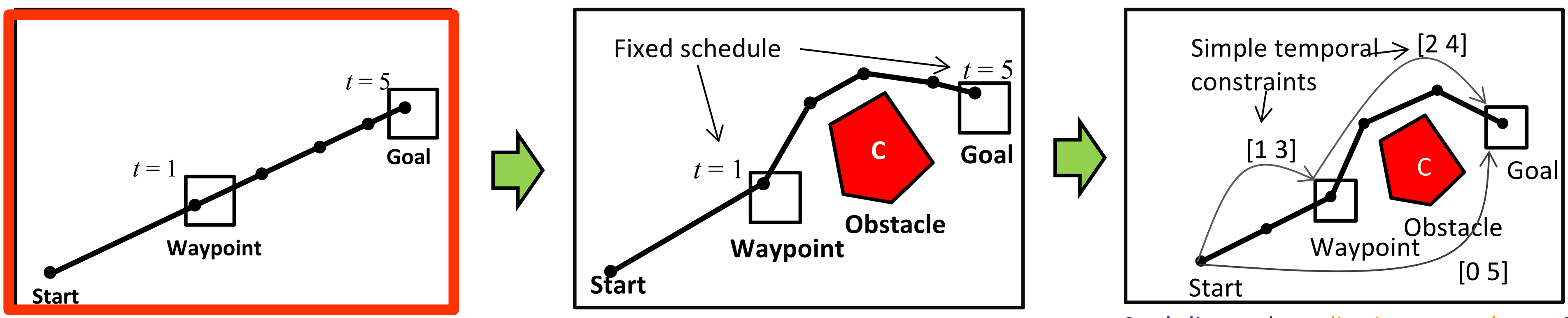
Outline

- A Risky Business
- Overview of Risk-bounded Planning
- Goal-directed Trajectory Planning
- Risk-bounded Trajectory Planning
  - Problem Statement
  - Elliptic Approximation
  - Iterative Risk Allocation (IRA)
  - Convex Risk Allocation (CRA)
    - Intuitions
    - Math



Approach: Solve increasingly expressive problems

(QSP)



Convex chance-constrained traj opt

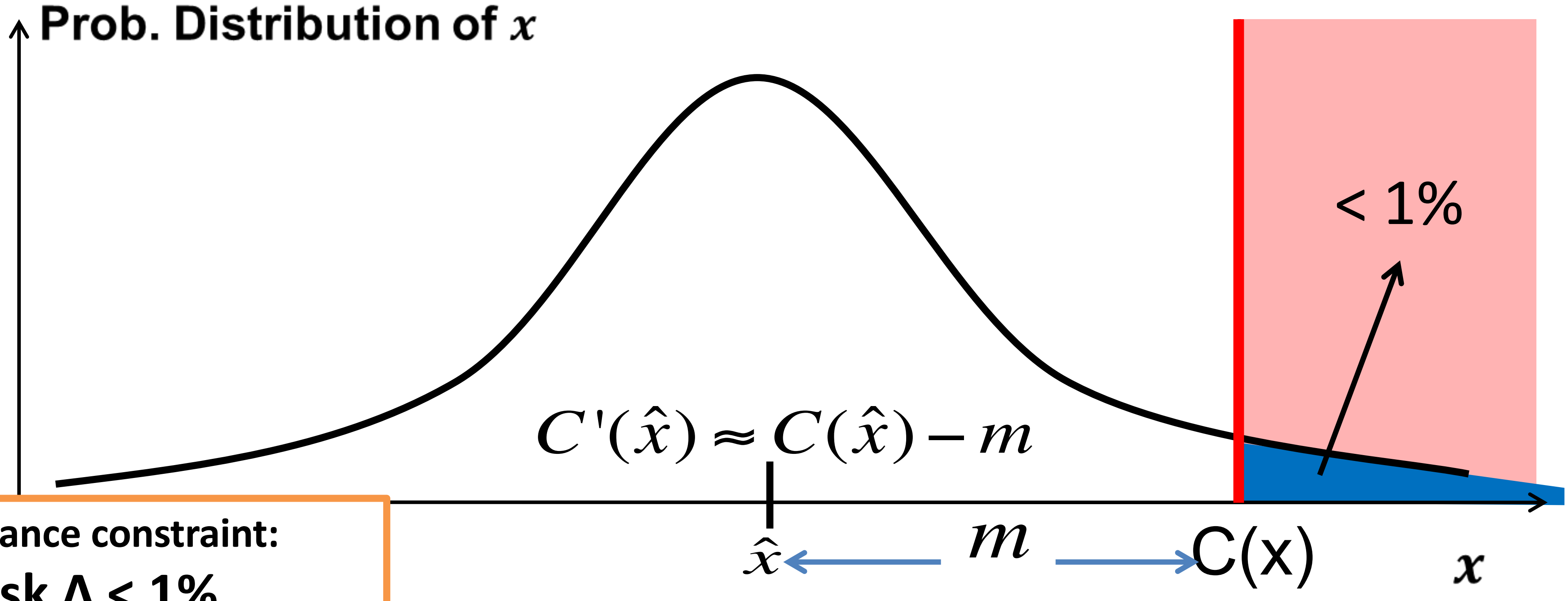
Non-convex, chance-constrained traj opt

Goal-directed qualitative state plan traj opt

Closed-form Encoding for Convex Problem

Risk-Allocation Overview: One Variable

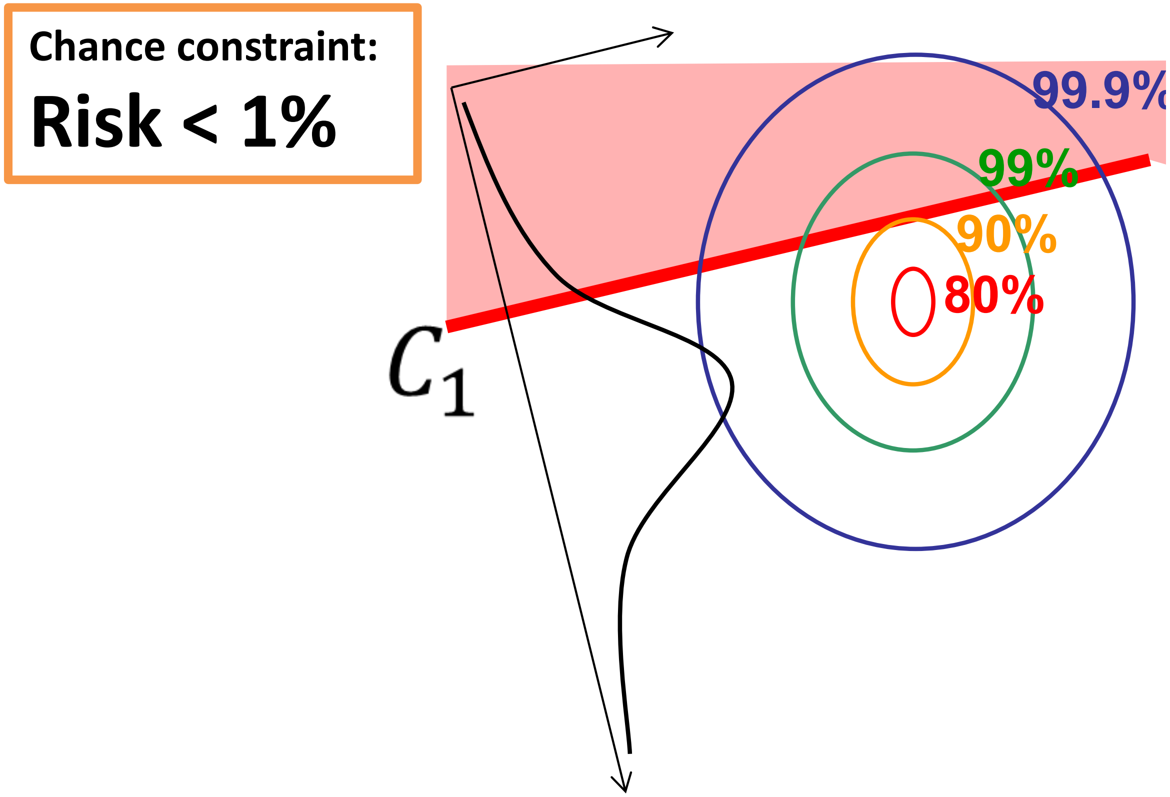
Idea 1: Given linear constraint C and normally distributed variable x, easy to solve by reformulating C to a deterministic constraint C' on $\hat{x}$.



Chance constraint:

Risk Δ < 1%

Risk-Allocation Overview: Many Variables



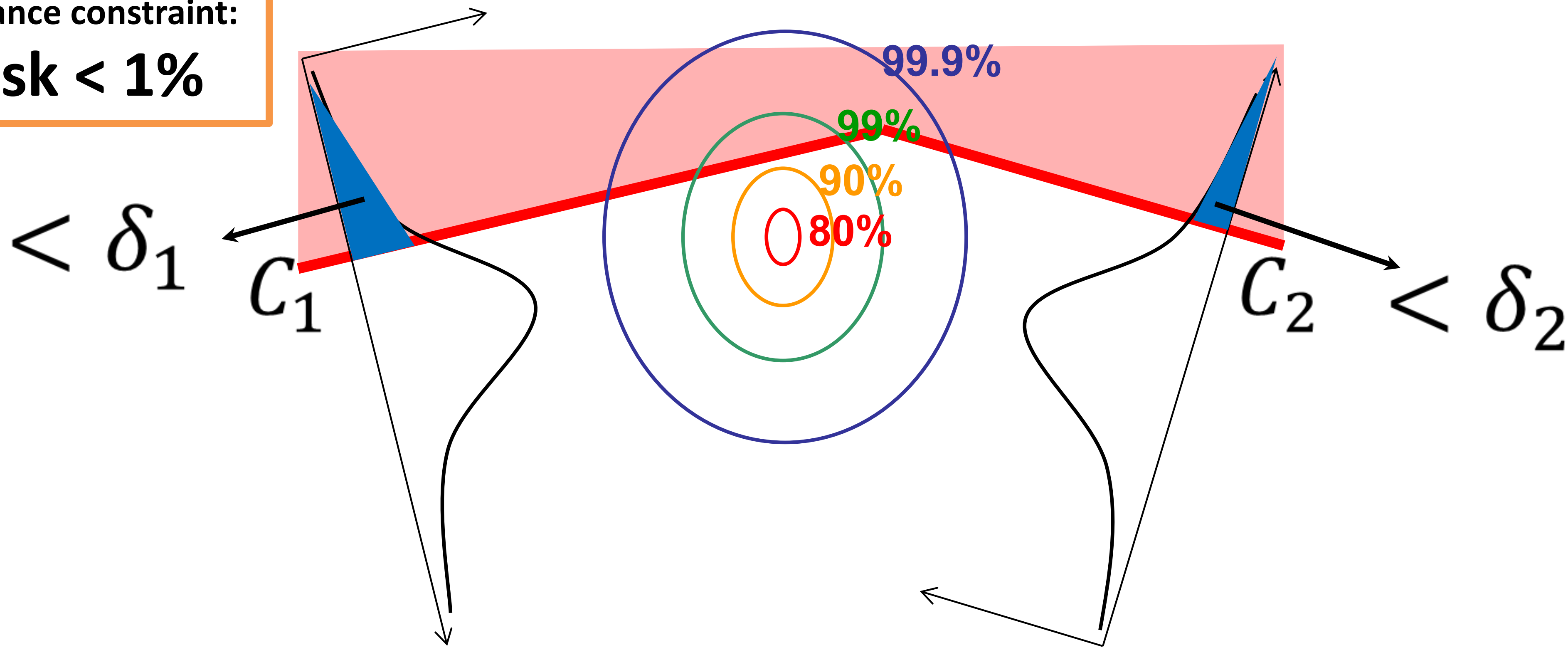
Idea 2: Generalize to a single constraint over an N-dimensional random variable,

by projecting its distribution onto the axis perpendicular to the constraint boundary.

Risk-Allocation Overview: Many Constraints

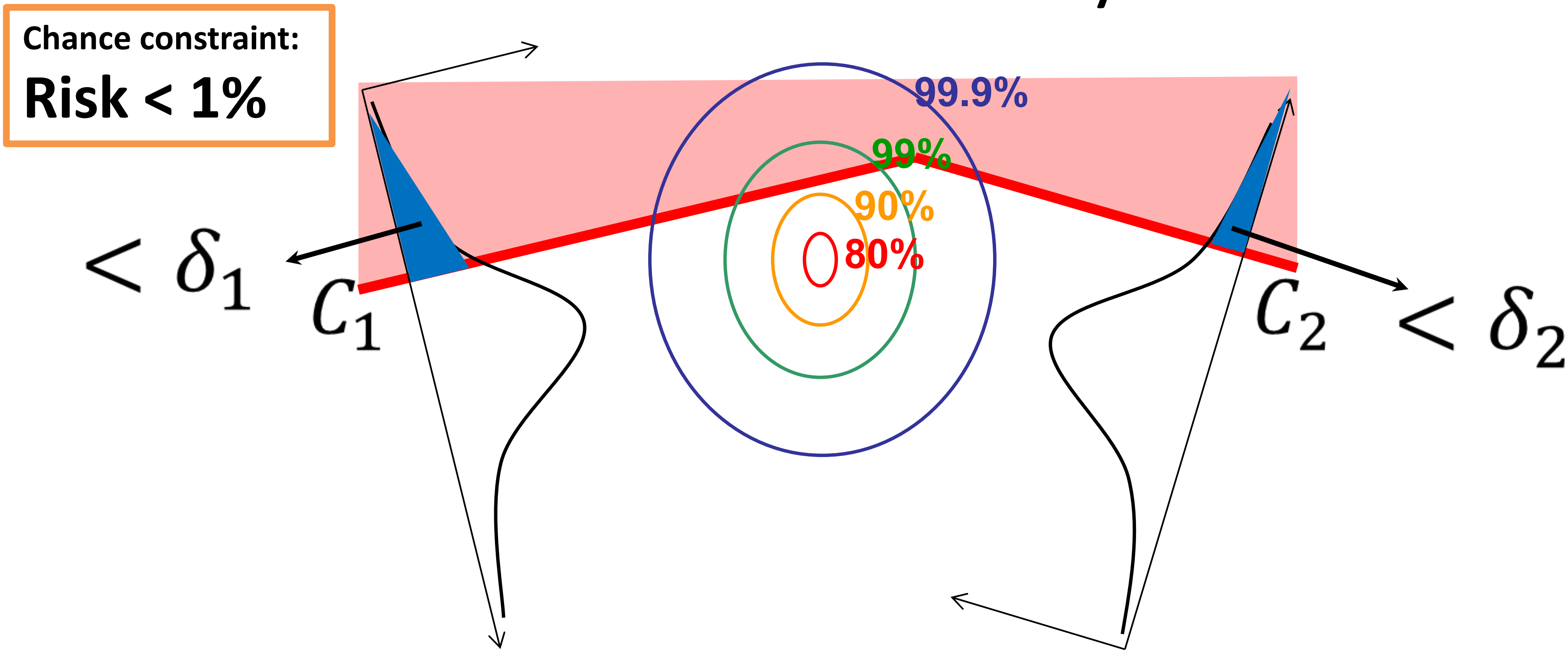
Chance constraint:

Risk < 1%



Idea 3: Generalize to a joint chance-constraint over multiple constraints C_1, C2, by distributing risk.

Risk-Allocation Overview: Many Constraints



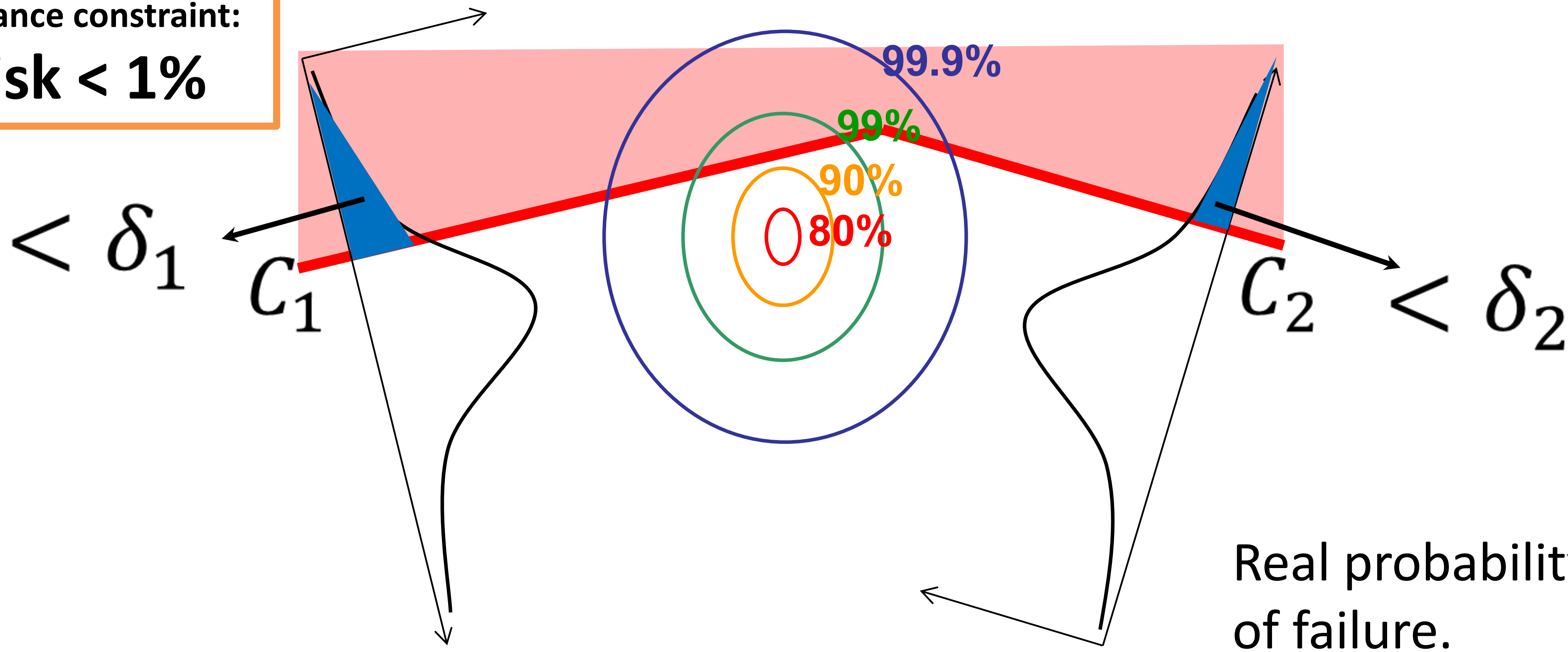
Find a solution such that:

1. Each constraint C_i takes less than δ_i risk, and
2. $\Sigma_i \, \delta_i \leq 1\%$

Note: this bound is derived from Boole's inequality.

Risk-Allocation Overview: Many Constraints

Chance constraint:
Risk < 1%



Real probability of failure.

Using Boole's inequality,

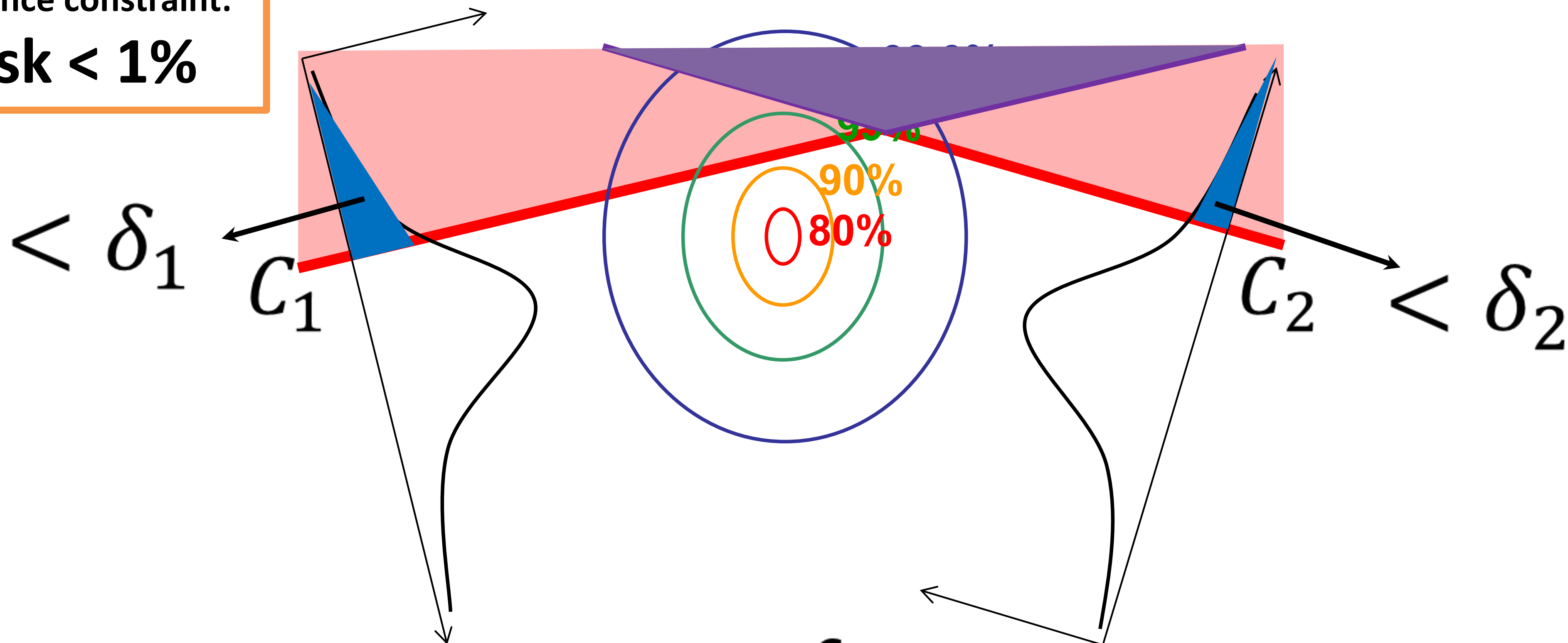
$$1\% \geq \Pr[F_1] + \Pr[F_2] \geq \Pr[F_1 \cup F_2]$$

where F_i is an event in which C_i is violated.

Risk-Allocation Overview: Conservatism

Chance constraint:

Risk < 1%



$$\text{Conservatism} = \int p(x)dx$$

Significantly less conservative than the elliptic approximation, especially in a high-dimensional problem.

Approach

- **Determinize problem through risk allocation**

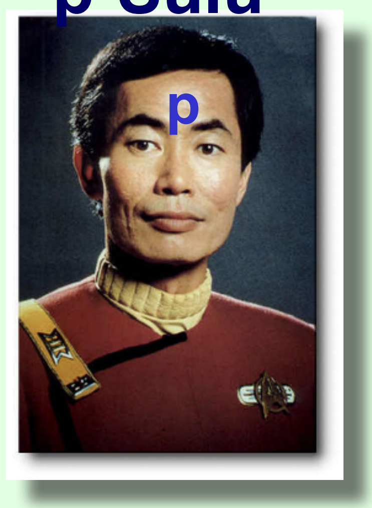
Input

Goal (cc-QSP) +

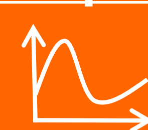
Stochastic Model

($X_{t+1} = AX_t + Bu_t + w_t$)

p-Sulu



Output



Control Sequence

Convex Programming Solver

Outline

- A Risky Business
- Overview of Risk-bounded Planning
- Goal-directed Trajectory Planning
- Risk-bounded Trajectory Planning
  - Problem Statement
  - Elliptic Approximation
  - Iterative Risk Allocation (IRA)
  - Convex Risk Allocation (CRA)
    - Intuitions
    - Math

Reformulation: Now Lets Do the Math!

$$\min_{u_{1:T} \in \mathbf{U}^T} J(u_{1:T})$$

$$s.t.$$

Stochastic dynamics

$$\bigwedge_{t=0}^{T-1} x_{t+1} = Ax_t + Bu_t + w_t$$

$$w_t \sim N(0, \Sigma_t)$$

$$x_0 \sim N(\bar{x}_0, \Sigma_{x,0})$$

Chance constraint

$$\Pr\left[\bigwedge_{t=1}^{T} \bigwedge_{i=1}^{N} h_t^{iT} x_t \leq g_t^i\right] \geq 1 - \Delta$$

Risk bound
(Upper bound of the probability of failure)
Assumption: $\Delta < 0.5$

Conversion of Joint Chance Constraint

Joint chance constraint

$$\Pr\left[\bigwedge_{t=0}^{T}\bigwedge_{i=0}^{N} h_t^{iT} x_t \le g_t^i\right] \ge 1-\Delta$$

Intractable

- Requires computation of an integral over a multivariate Gaussian.

1

A set of individual chance constraints.

- involves univariate Gaussian distribution.

2

A set of deterministic state constraint.

Decomposition of Joint Chance Constraint

Joint chance constraint

$$\Pr\left[\bigwedge_{t=0}^{T}\bigwedge_{i=0}^{N} h_t^{iT} x_t \le g_t^i\right] \ge 1-\Delta$$

1

Using Boole's inequality (union bound)

$$\Pr[A \cup B] \le \Pr[A] + \Pr[B]$$

Where A and B denote constraint failures

Decomposition of Joint Chance Constraint

Joint chance constraint

$$\Pr\left[\bigwedge_{t=0}^{T}\bigwedge_{i=0}^{N} h_t^{iT} x_t \le g_t^i\right] \ge 1 - \Delta$$

Constant

Upper bound of the probability of violating any constraints over the planning horizon

is implied by:

Individual chance constraints

$$\bigwedge_{t=1}^{T}\bigwedge_{i=1}^{N}\left(\Pr\left[h_t^{iT} x_t \le g_t^i\right] \ge 1 - \delta_t^i\right)$$

Decision Variable

$$\sum_{t,i} \delta_t^i \le \Delta$$

Upper bound of the probability of violating ith constraint at time t

$$\bigwedge_{t=1}^{T}\bigwedge_{i=1}^{N} \delta_t^i \ge 0$$

Risk allocation:

$$\boldsymbol{\delta} = \left[\delta_1^1, \delta_1^2 \cdots \delta_T^N\right]$$

1

Decomposition of Joint Chance Constraint

$$\min_{\boldsymbol{\delta}} \min_{u_{1:T} \in \mathbf{U}^T} J(U)$$

Risk allocation optimization

$$s.t.$$

$$\bigwedge_{t=0}^{T-1} x_{t+1} = Ax_t + Bu_t + w_t$$

$$w_t \sim N(0, \Sigma_t)$$

$$x_0 \sim N(\bar{x}_0, \Sigma_{x,0})$$

Individual chance constraints

$$\bigwedge_t \bigwedge_i \Pr\left[h_t^{iT} x_t \le g_t^i\right] \ge 1 - \delta_t^i$$

$$\sum_{t,i} \delta_t^i \le \Delta$$

Joint chance constraint

$$\cancel{\Pr\left[\bigwedge_{t=0}^{T} \bigwedge_{i=0}^{N} h_t^{iT} x_t \le g_t^i\right] \ge 1 - \Delta}$$

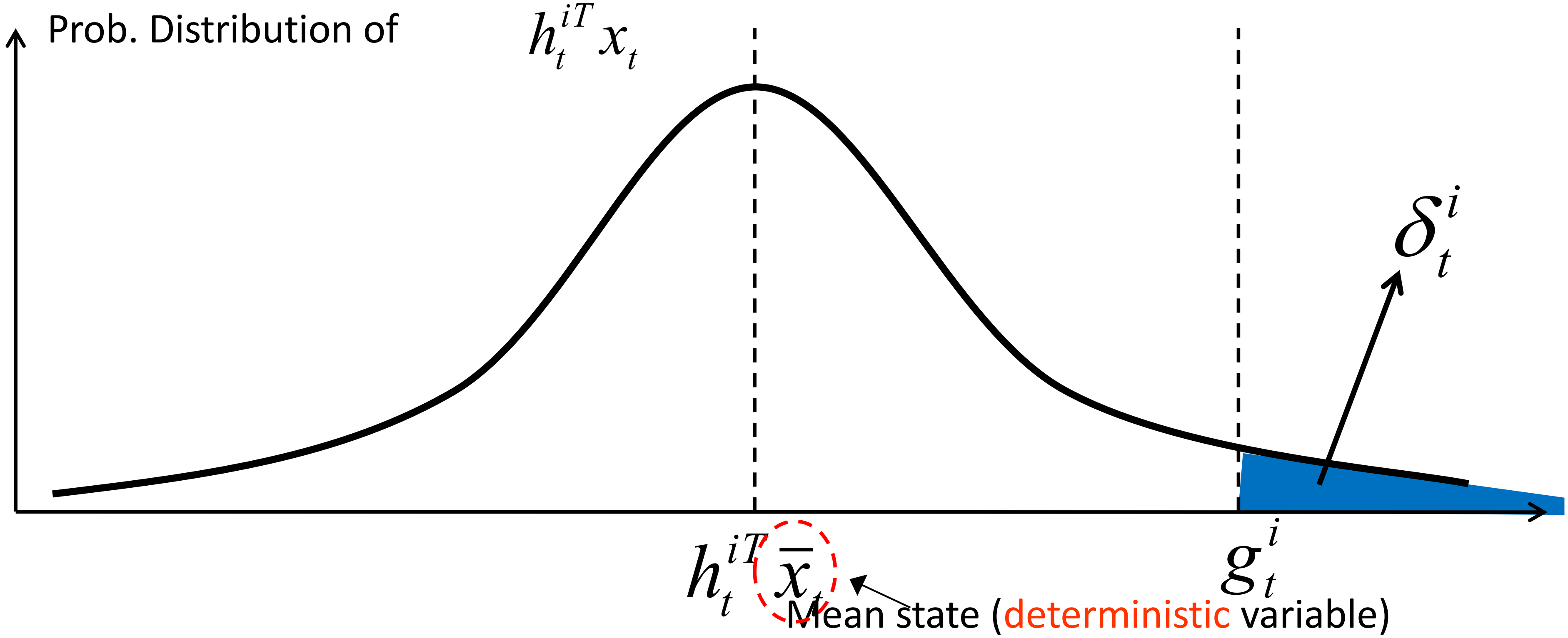
2

Conversion to Deterministic Constraint

Chance constraint

$$\Pr\left[\underline{h_t^{iT} x_t} \le g_t^i\right] \ge 1 - \delta_t^i$$

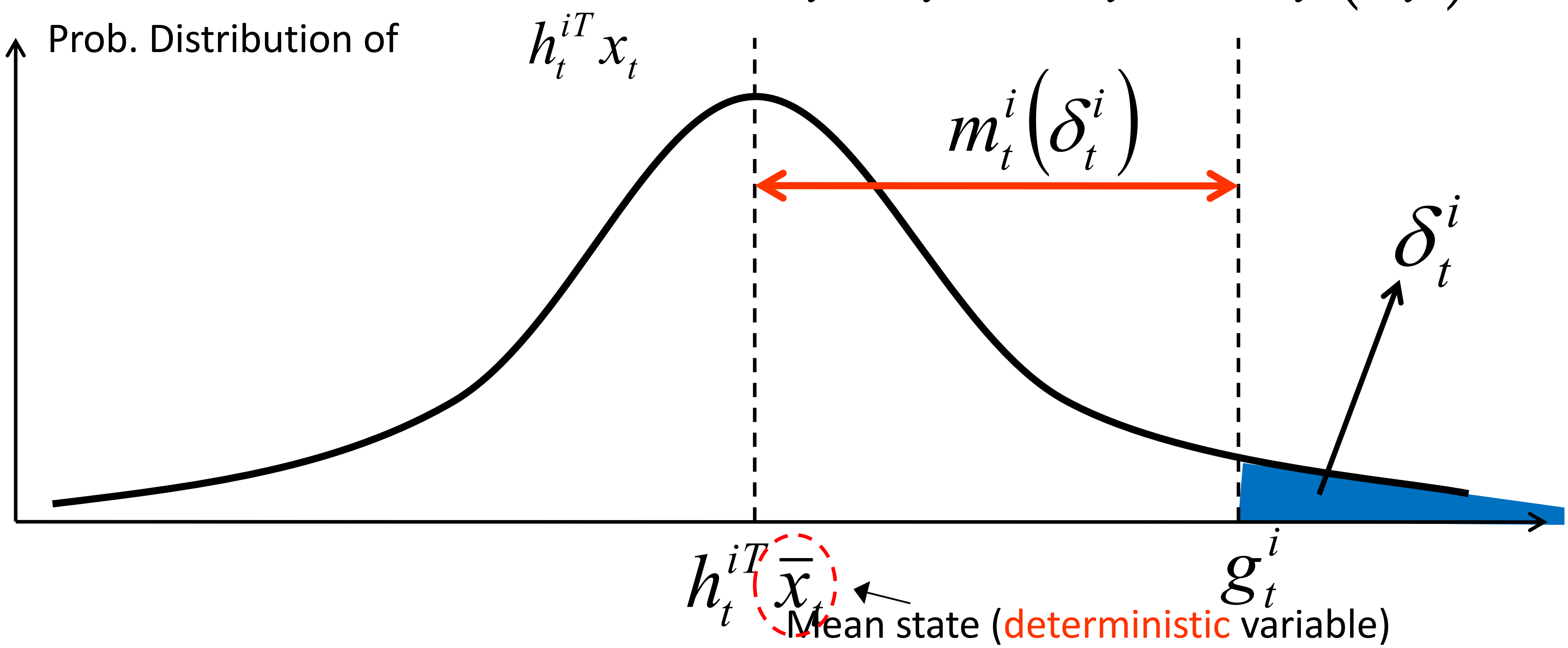
Univariate Gaussian distribution



Conversion to Deterministic Constraint

Chance constraint $$\Pr\left[h_t^{iT} x_t \le g_t^i\right] \ge 1 - \delta_t^i$$

Deterministic constraint $$\Leftrightarrow h_t^{iT} \bar{x}_t \le g_t^i - m_t^i\left(\delta_t^i\right)$$



2 Conversion to Deterministic Constraint

Chance constraint

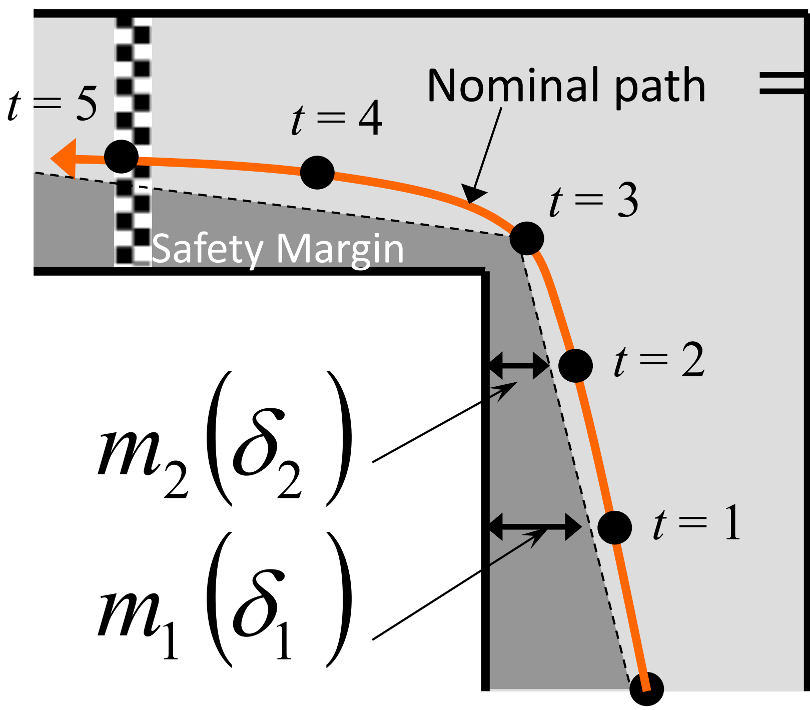
$$\Pr\left[h_t^{iT} x_t \le g_t^i\right] \ge 1 - \delta_t^i$$

Deterministic constraint

$$\Leftrightarrow h_t^{iT} \bar{x}_t \le g_t^i - \underline{m_t^i\left(\delta_t^i\right)}$$

Mean state

Safety Margin



= Mean state

[Charnes et. al. 1959]

2 Conversion to Deterministic Constraint

Chance constraint

$$\Pr\left[h_t^{iT} x_t \le g_t^i\right] \ge 1 - \delta_t^i$$

Deterministic constraint

$$\Leftrightarrow h_t^{iT} \bar{x}_t \le g_t^i - m_t^i\left(\delta_t^i\right)$$

where

$$m_t^i\left(\delta_t^i\right) = -\sqrt{2 h_t^{iT} \Sigma_{x,t} h_t^i}\,\mathrm{erf}^{-1}\left(2\delta_t^i - 1\right)$$

(Inverse of cdf of Gaussian)

$$x_t \sim N(\bar{x}_t, \Sigma_{x,t})$$

$$\Sigma_{x,t} = \sum_{n=0}^{t-1} A^n \Sigma_w (A^n)^T + \Sigma_{x,0}$$

[Charnes et. al. 1959]

2

Conversion to Deterministic Constraint

$$\min_{\boldsymbol{\delta}} \min_{u_{1:T} \in \mathbf{U}^T} J(u_{1:T})$$

$$s.t.$$

$$\bigwedge_{t=0}^{T-1} x_{t+1} = Ax_t + Bu_t + w_t$$

$$w_t \sim N(0, \Sigma_t)$$

$$x_0 \sim N(\bar{x}_0, \Sigma_{x,0})$$

Individual chance constraints

$$\bigwedge_{t=1}^{T} \bigwedge_{i=1}^{I} \Pr\left[h_t^{iT} x_t \le g_t^i\right] \ge 1 - \delta_t^i$$

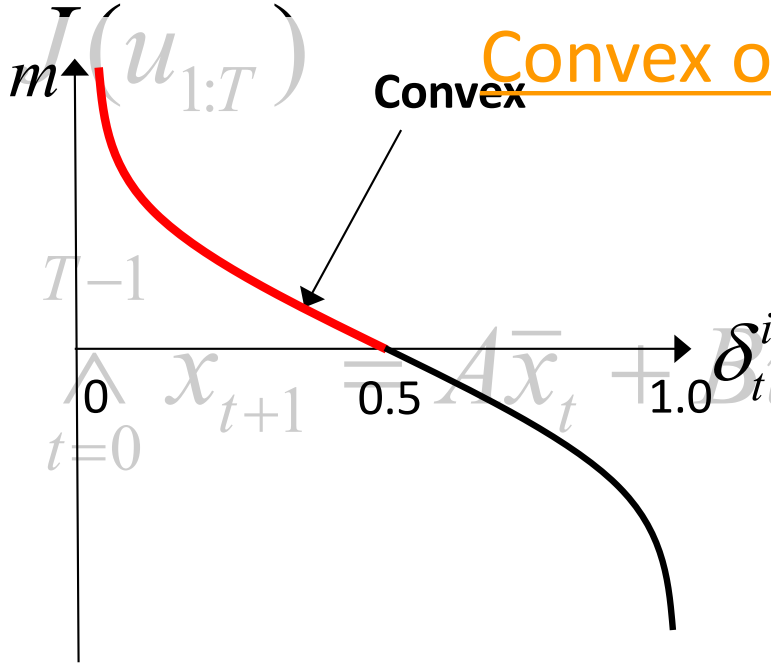
$$\sum_{t,i} \delta_t^i \le \Delta$$

Conversion to Deterministic Constraint

$$\min_{\delta} \min_{u_{1:T} \in \mathbf{U}^T} J(u_{1:T})$$

Convex optimization

$$s.t. \quad \bigwedge_{t=0}^{T-1} x_{t+1} = A\bar{x}_t + Bu_t$$

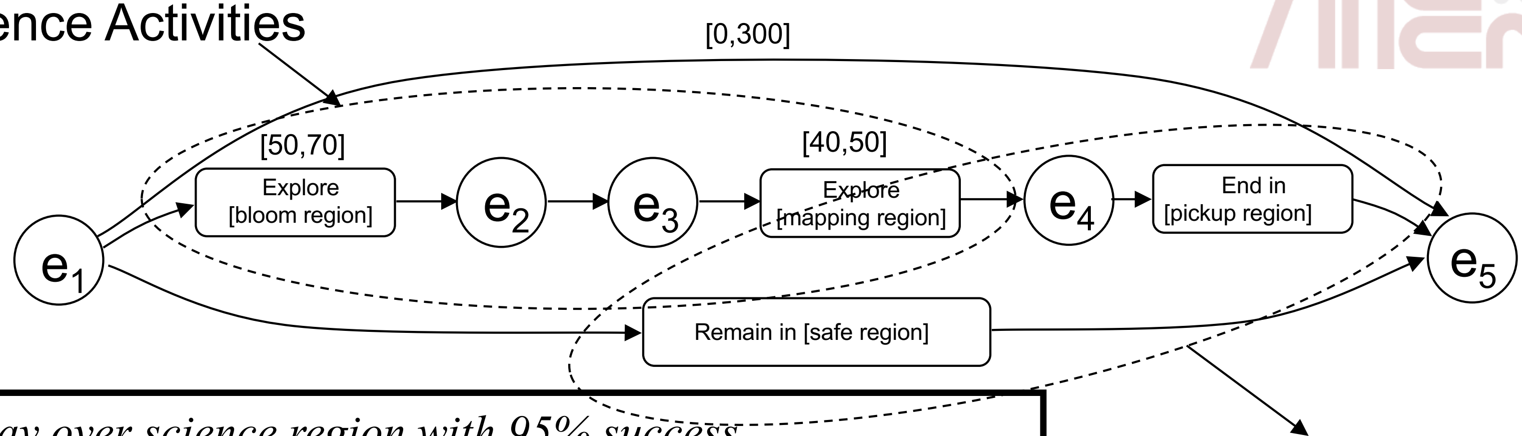


$$\bigwedge_{t=1}^{T} \bigwedge_{i=1}^{I} h_t^{iT} \bar{x}_t \le g_t^i - m_t^i\left(\delta_t^i\right)$$

Convex if $\delta < 0.5$

$$\sum_{t,i} \delta_t^i \le \Delta$$

1. Science Activities

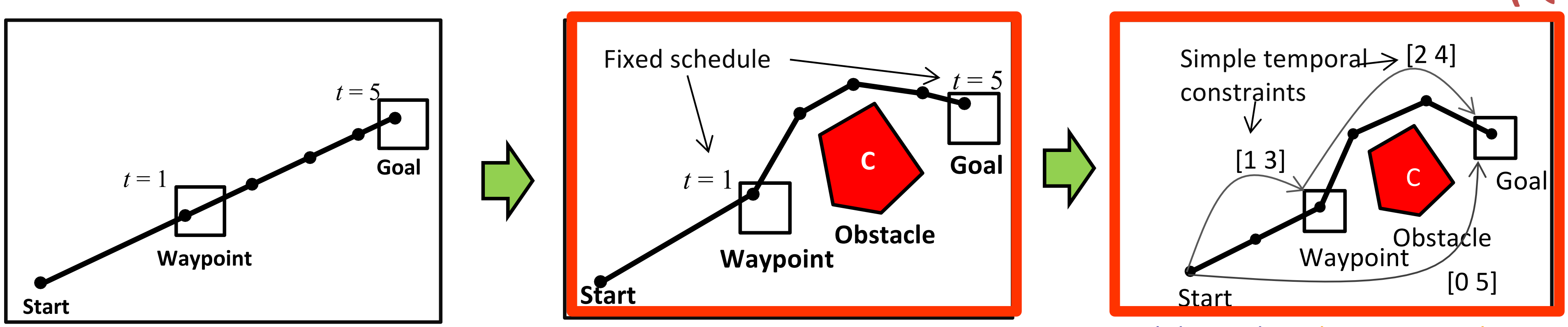


"Stay over science region with 95% success.
Avoid collision and achieve pickup with 99.9% success."

2. Safety Activities

Approach: Solve increasingly expressive problems

(QSP)



Convex chance-constrained traj opt

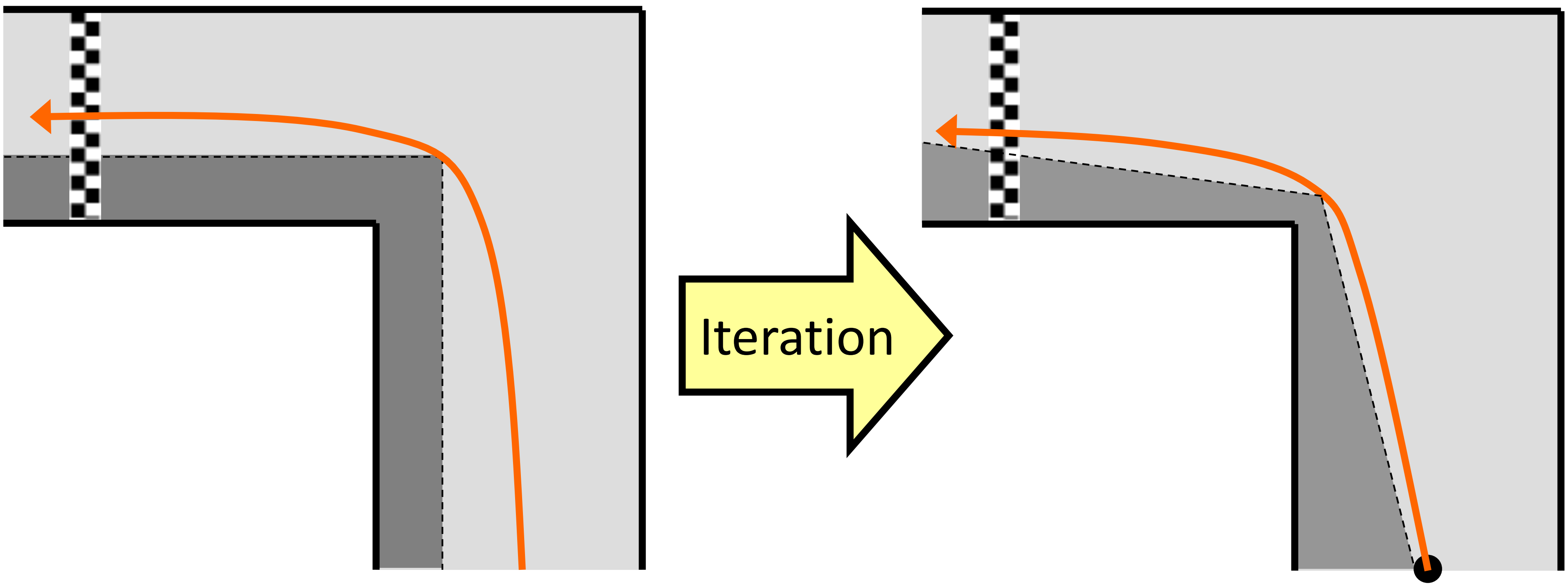
Non-convex, chance-constrained traj opt

Goal-directed qualitative state plan traj opt

B&B and risk allocation over
stochastic disjunctive linear / temporal constraints

Summary: Risk Allocation

$$\bar{J}^*(\boldsymbol{\delta}_0) \geq \bar{J}^*(\boldsymbol{\delta}_1) \geq \bar{J}^*(\boldsymbol{\delta}_2)\cdots$$



1. IRA: reallocates risk manually.
2. CRA,NRA: standard solver reallocates risk.

Now you know …

- That autonomous systems are risky,
  and this risk must be managed.
- How to specify goal-directed motion planning problems as qualitative state plans (QSP).
- How to encode QSP motion planning as constraint optimization problems.
- When planning to maximize utility
  under bounded risk makes sense.
- How to use risk allocation solve trajectory and probabilistic planning problems and scheduling problems.
- How to perform risk allocation iteratively and in closed form.