Risk-bounded Programming on Continuous State

Prof. Brian Williams

March 30th, 2016

Cognitive Robotics (16.412J / 6.834J)

photo courtesy MIT News

Assignments

Today: Risk-bounded Motion Planning

- 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, December 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.

After Advanced Lectures: Risk-bounded Scheduling

- C. Fang, P. Yu, and B. C. Williams, "Chance-constrained Probabilistic Simple Temporal Problems," AAAI, Montreal, CN, 2014.
- A. Wang and B. C. Williams, "Chance-constrained Scheduling via Conflict-directed Risk Allocation," *AAAI*, Austin, TX, January, 2015.

After Advanced Lectures: Risk-bounded Probabilistic Activity Planning

- Santana, P., Thiébaux, S., Williams, B.C., "RAO*: an Algorithm for Chance-Constrained POMDP's," AAAI, Phoenix, AZ, February 2016.

Homework:

- Changing Pset order; different from syllabus.
- IRA pset soon (next 1-2 weeks)
- Let Steve know what times work for advanced lecture dry runs

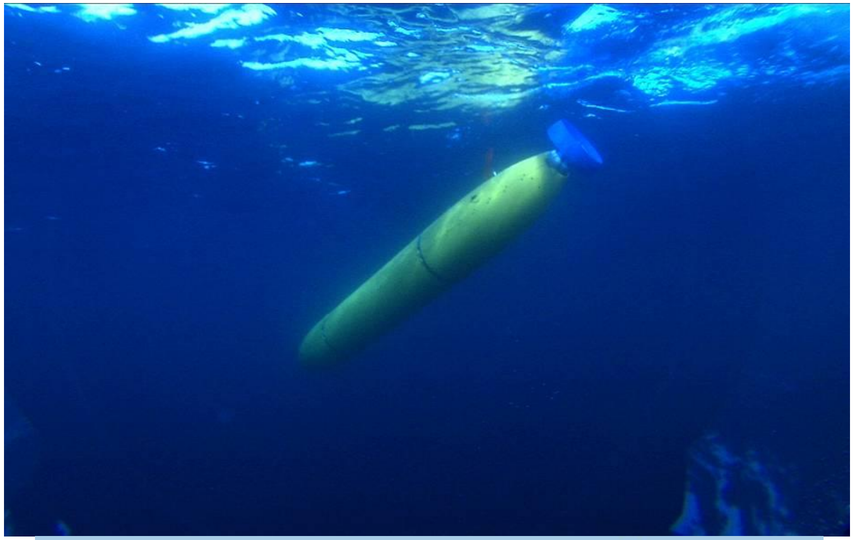
Key takeaways

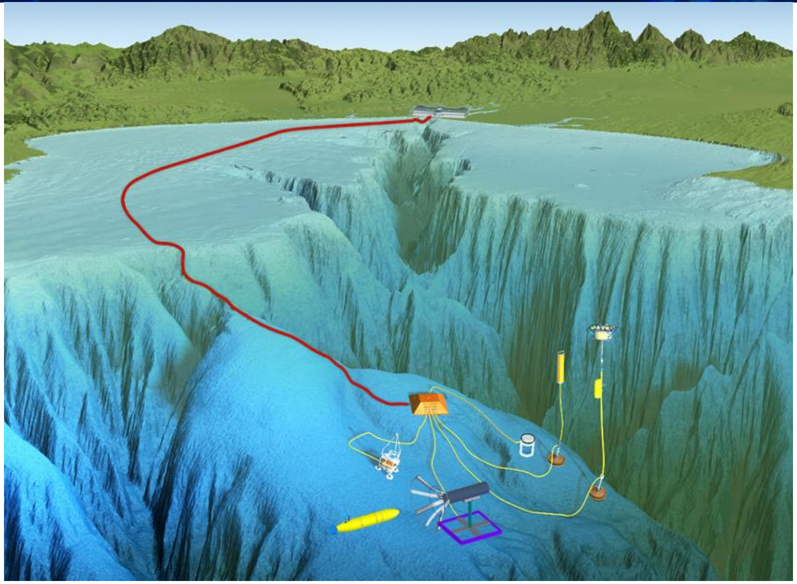
- Maximizing utility under bounded risk makes sense.
- Risk allocation can help us solve.

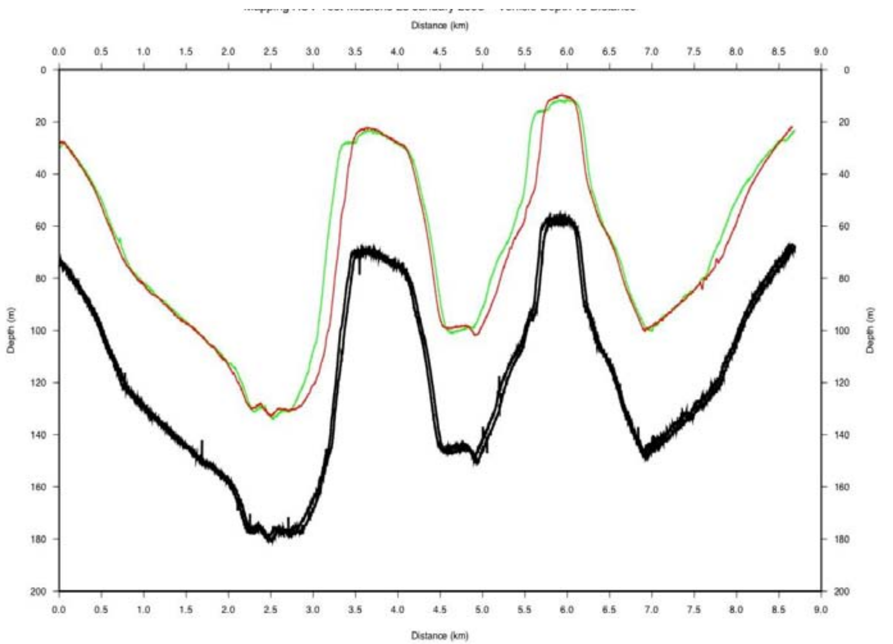
Outline

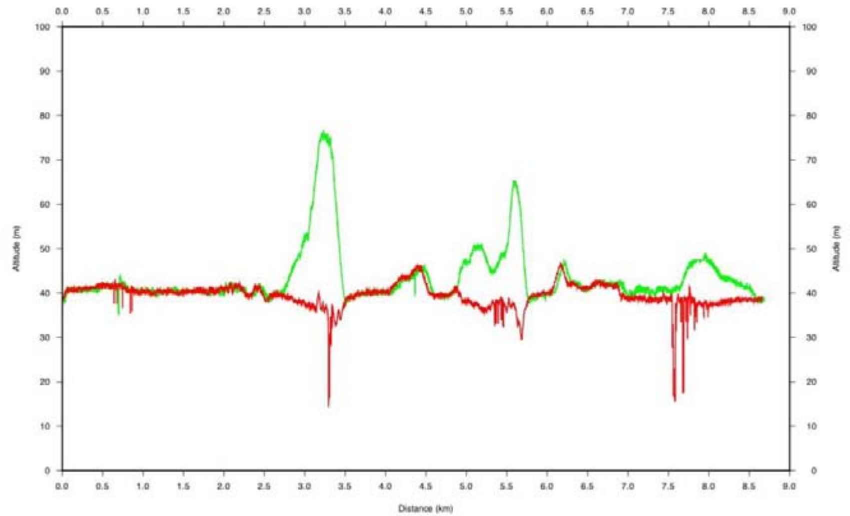
- Review
- Risk-aware Trajectory Planning
- Iterative Risk Allocation (IRA)
- Generalizing to Risk-aware Systems
- Convex Risk Allocation (CRA)

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









Dynamic Execution of State Plans

00:00 Go to x_1,y_1
00:20 Go to x_2,y_2
00:40 Go to x_3,y_3
…
04:10 Go to x_n,y_n

Command script

Commands

Plant

Leaute & Williams, AAAI 05

Dynamic Execution of State Plans

"Explore mapping region for at least 100s, then explore bloom region for at least 50s, then return to pickup region. Avoid obstacles at all times"

State Plan

Sulu
Model-based Executive

Dynamics
+
Constraints

Observations

Commands

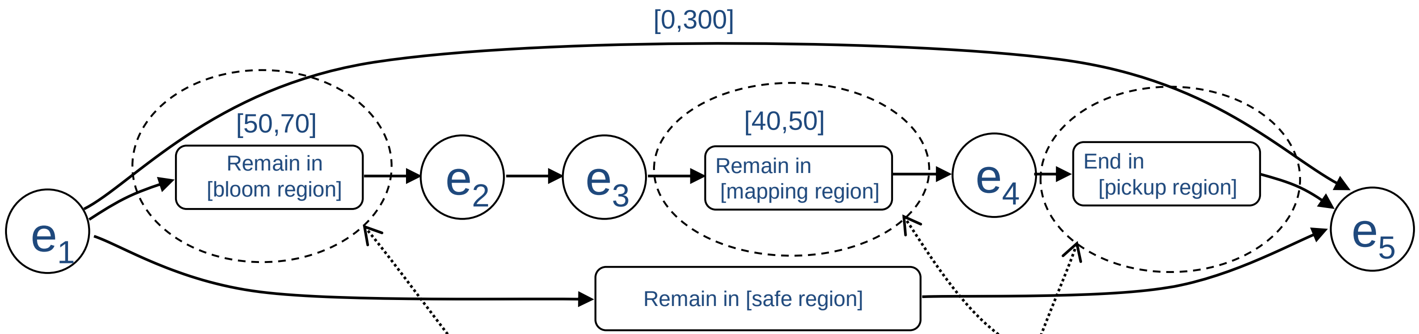
Plant

Optimal

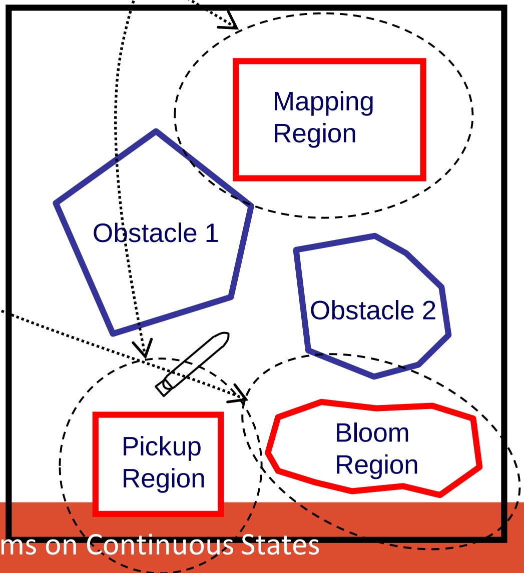
Leaute & Williams, AAAI 05

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 seconds. Afterwards, explore mapping region for between 40s and 50s. End in the pickup region. Avoid obstacles at all times. Complete the mission within 300s"



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]

Frame Planning as a Mathematical Program

$$\min_{\mathbf{x}_{1:N},\mathbf{u}_{1:N}} J(\mathbf{x}_1 \cdots \mathbf{x}_N, \mathbf{u}_1 \cdots \mathbf{u}_N) \; \underline{+ f(\mathbf{x}_N)}$$

Cost function

Cost-to-go function

$s.t.$

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

Dynamics

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

Spatial constraints

$$\mathbf{x}_0 = \mathbf{x}_{\text{start}}$$

Initial position and velocity

$$\cancel{\mathbf{x}_N = \mathbf{x}_{\text{goal}}}$$

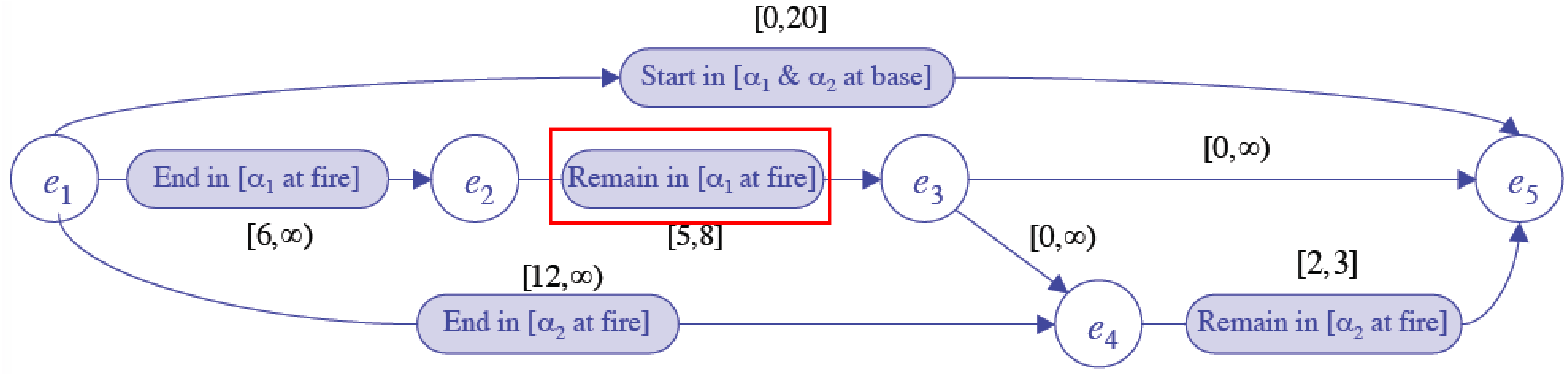
Goal position and velocity

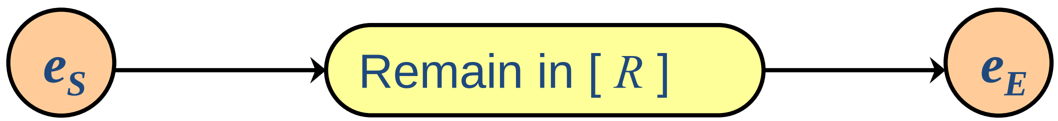
$$-\mathbf{u}_{\max} \le \mathbf{u}_k \le \mathbf{u}_{\max} \quad (k = 0,1,\cdots N-1)$$

Thrust limits

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

Encode “Remain In” Constraints, . . .





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

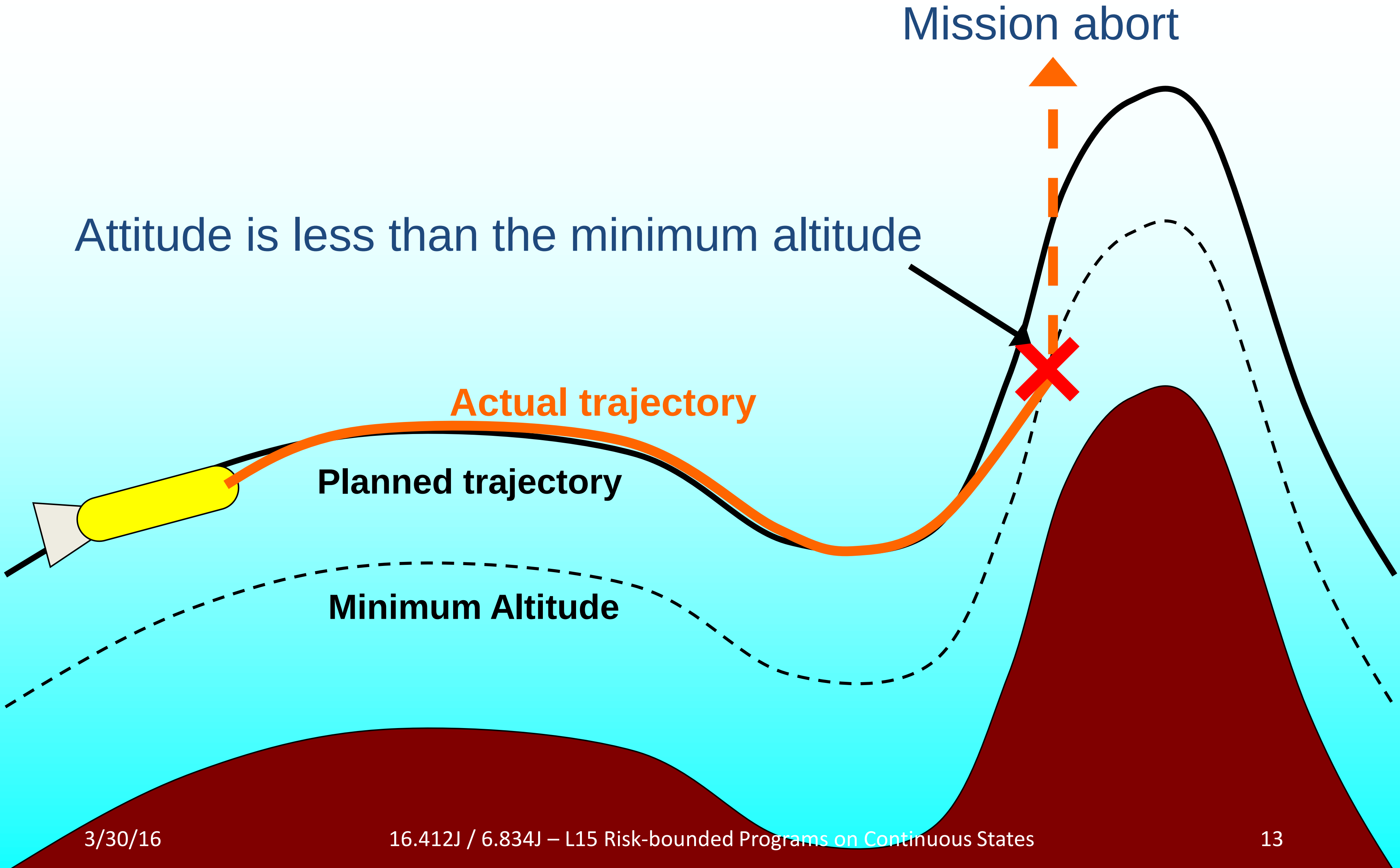
•Thomas Léauté, "*Coordinating Agile Systems through the Model-based Execution of Temporal Plans*, " S. M. Thesis, Massachusetts Institute of Technology, August 2005.
•Thomas Léauté, Brian Williams, “Coordinating Agile Systems Through the Model-based Execution of Temporal Plans," *Proceedings of the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

Outline

- Review
- Risk-aware Trajectory Planning
- Iterative Risk Allocation (IRA)
- Generalizing to Risk-aware Systems
- Convex Risk Allocation (CRA)

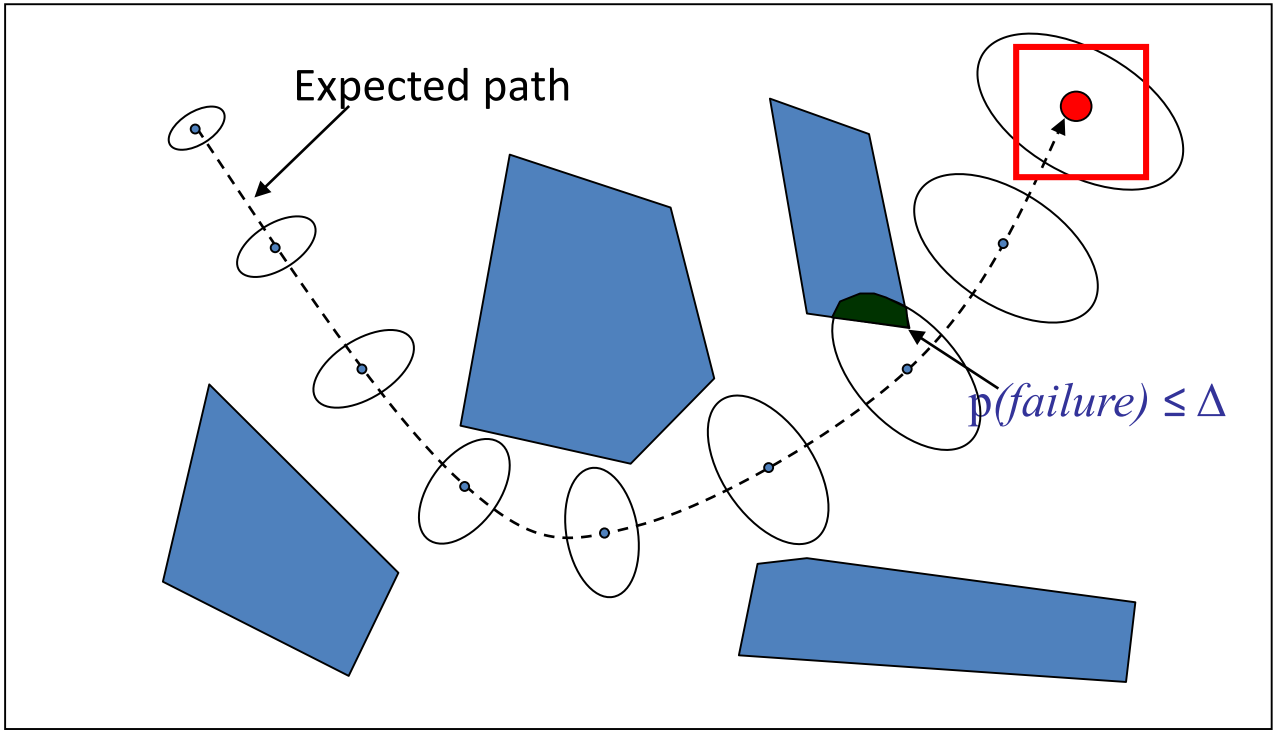

>$1M

Issue: Frequent Mission Aborts



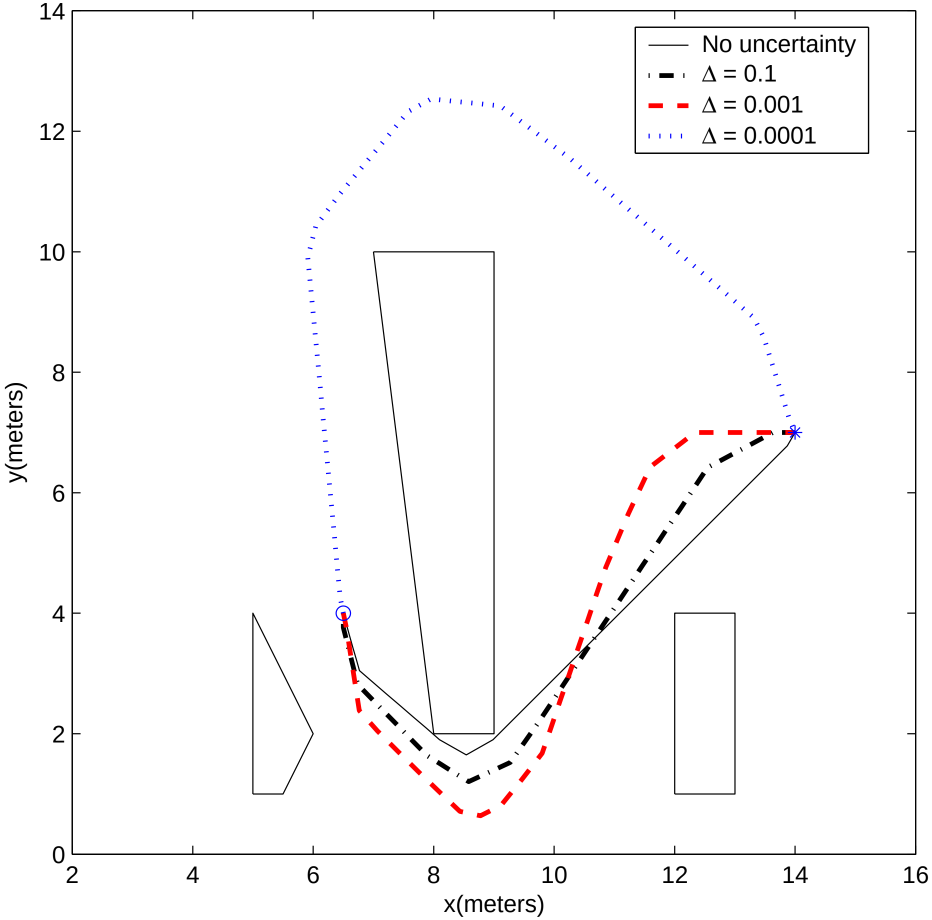
Chanced Constrained, Robust Path Planning

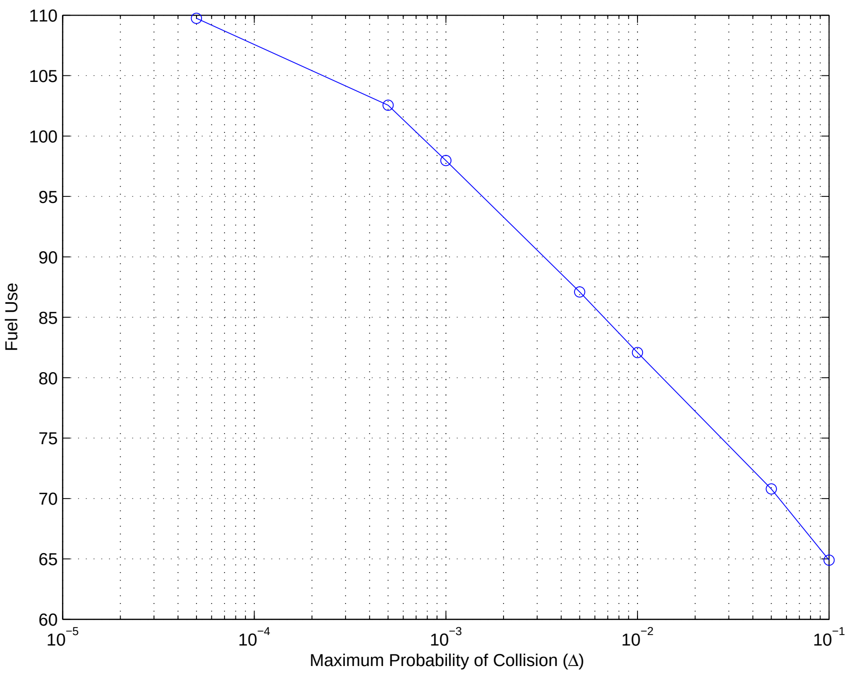
- "Plan optimal path to goal such that p*(failure)* ≤ Δ."



Risk – Performance Tradeoff

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



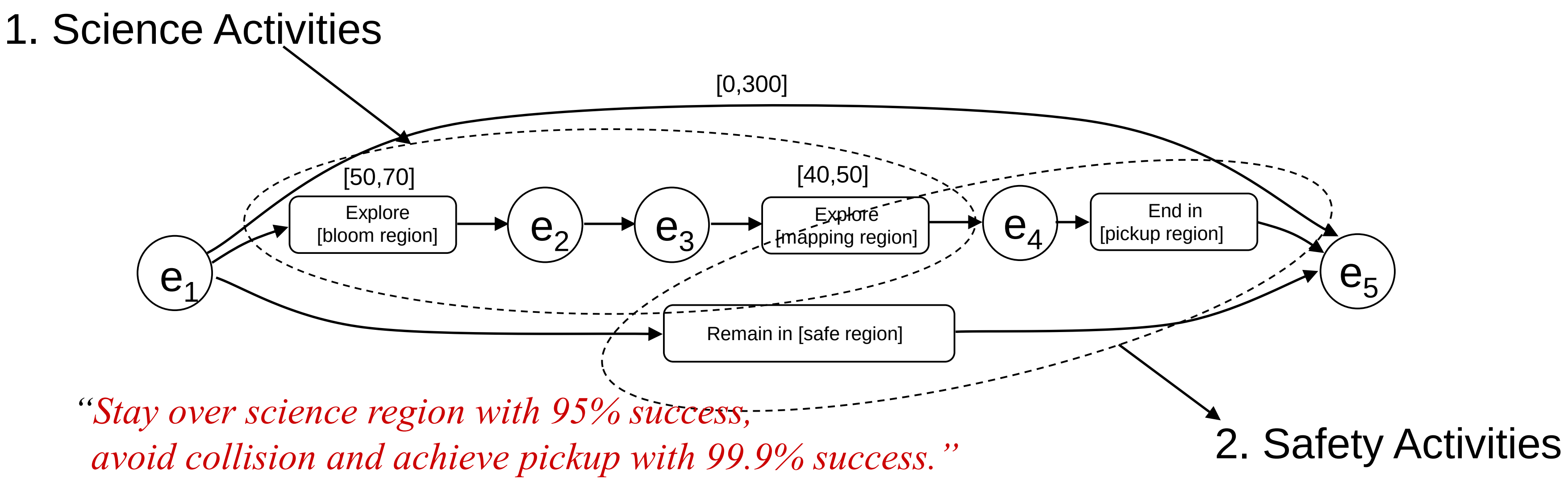


Method: Uniform Risk Allocation [Blackmore, PhD]

Goal-directed, Risk-bounded Planning

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):

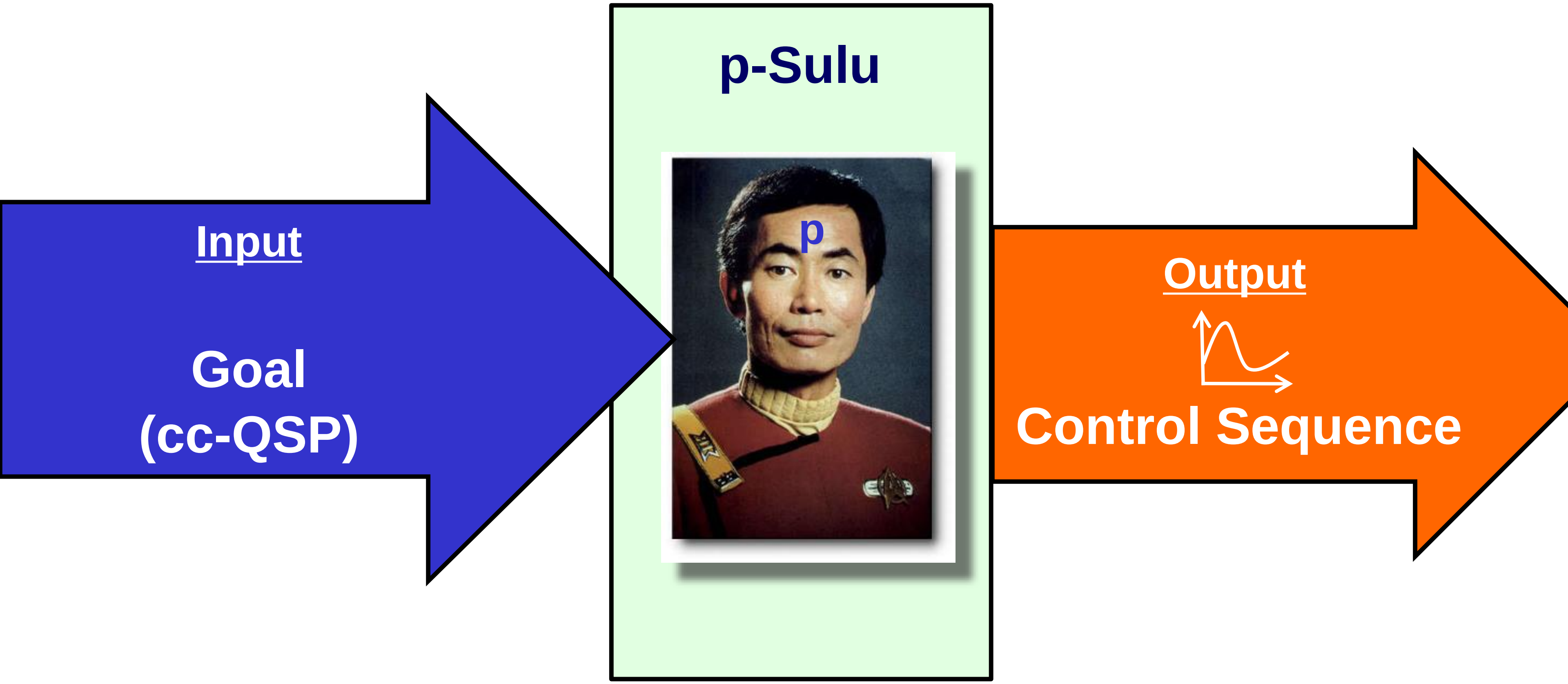
1. *p(* Remain in [bloom region] *fails* OR Remain in [mapping region] *fails*) < 5%.

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

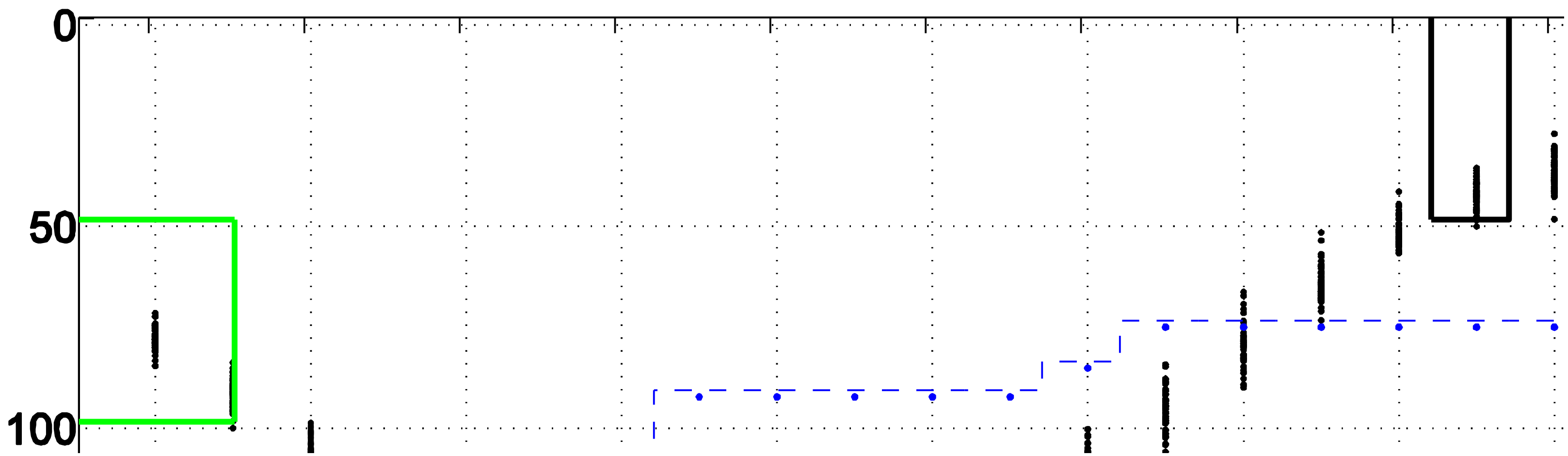
Instance of Chance-constrained Programming.

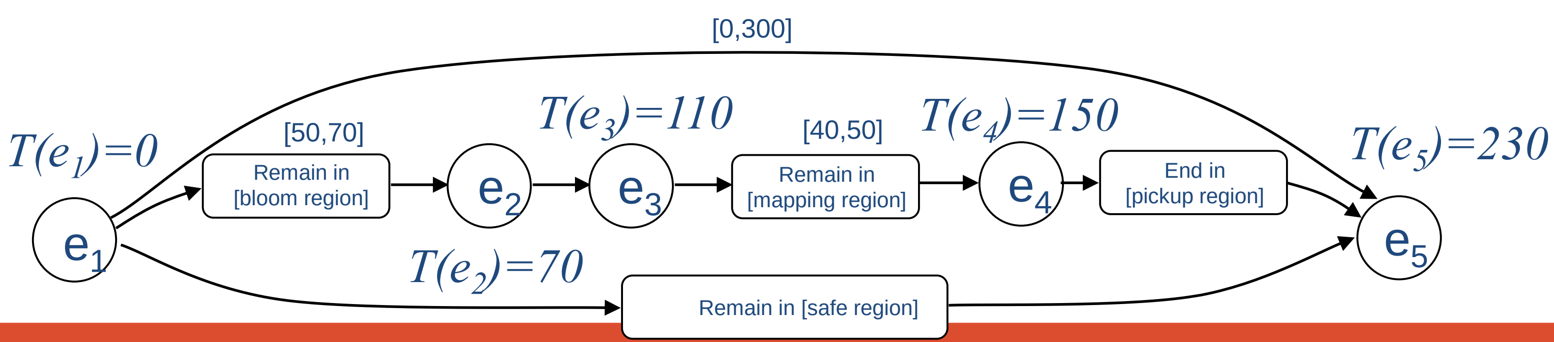
Input and Output

- **p-Sulu**: Probabilistic Sulu (plan executive)

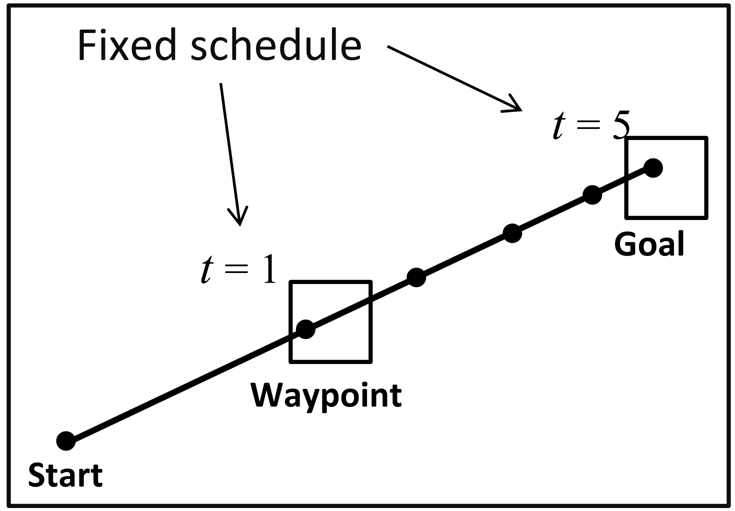


Example Execution

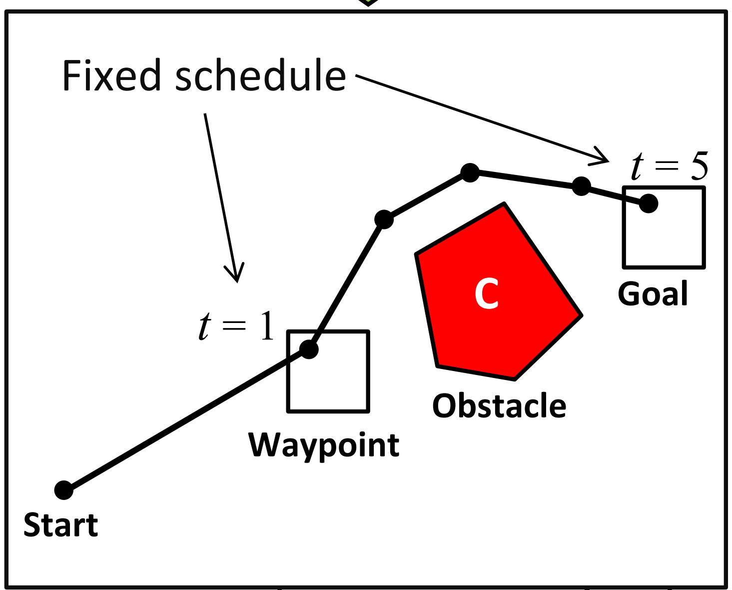




Problems



Convex chance-constrained traj opt



Non-convex, chance-constrained traj opt

Flexible schedule (QSP)

Deterministic Finite-Horizon Optimal Control

$$\min_{u_{1:T} \in \mathbf{U}^T} \underbrace{J(u_{1:T})}_{\text{Convex function}}$$

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

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

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

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

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

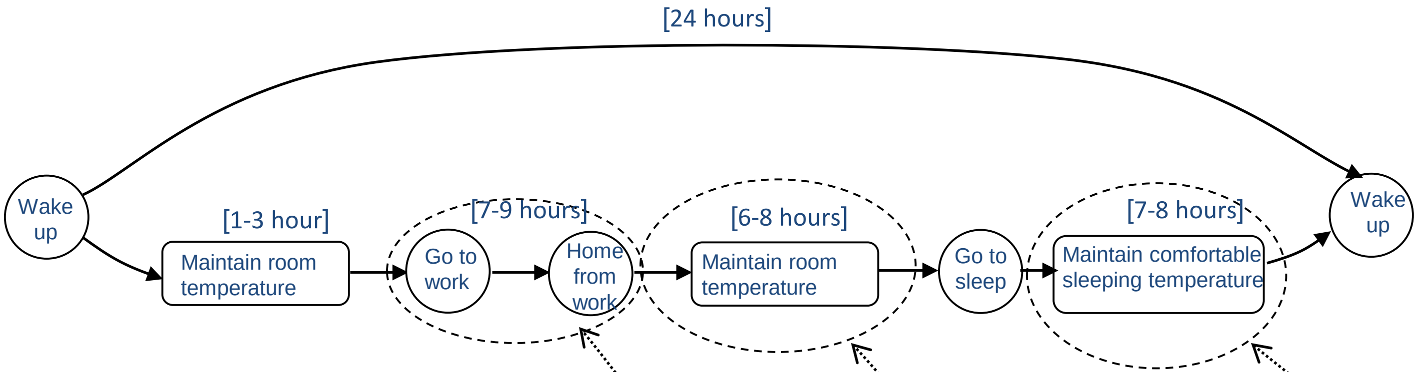
Example: Connected Sustainable Home

joint w F. Casalegno & B. Mitchell, MIT Mobile Experience Lab

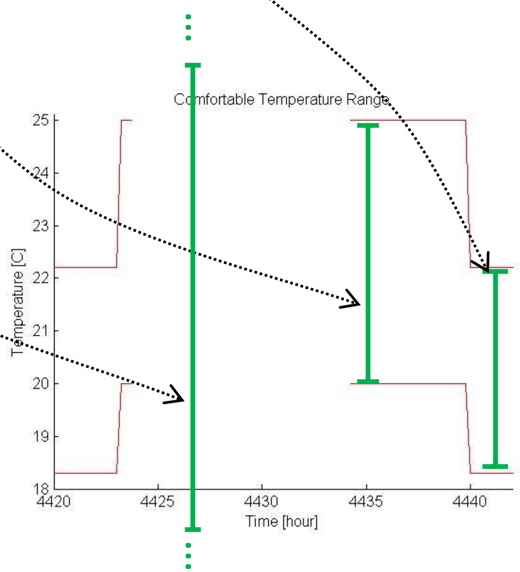
- Goal: Optimally control HVAC, window opacity, washer and dryer, e-car.
- Objective: Minimize energy cost.

Qualitative State Plan (QSP)

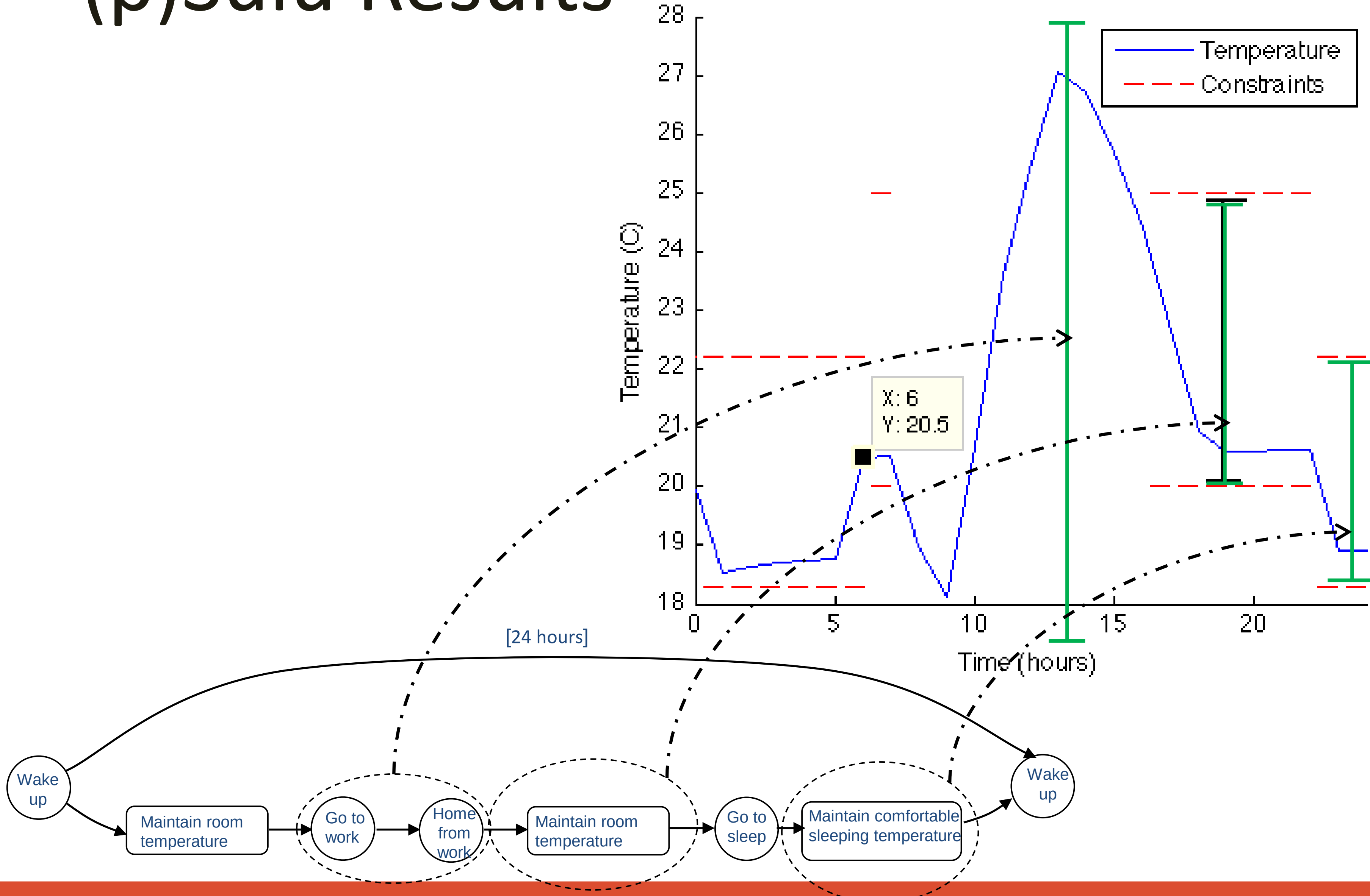
Sulu [Leaute & Williams, AAAI05]



"Maintain room temperature after waking up until I go to work. No temperature constraints while I'm at work, but when I get home, maintain room temperature until I go to sleep. Maintain a comfortable sleeping temperature while I sleep."



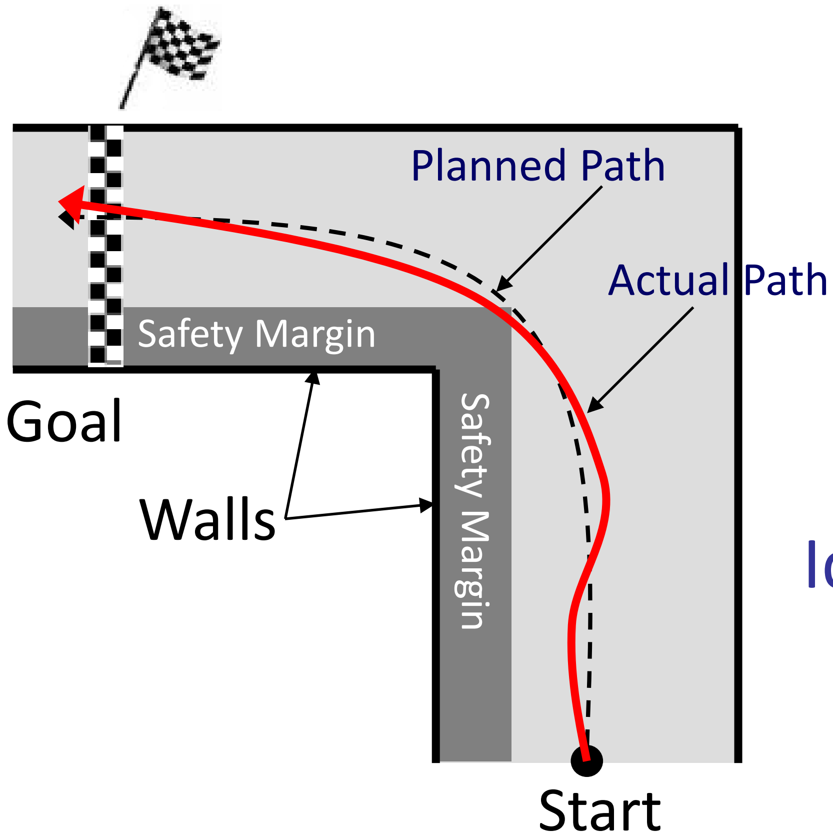
(p)Sulu Results

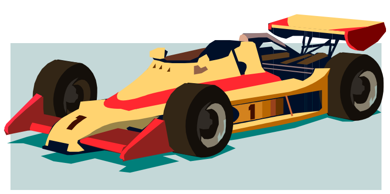


Outline

- Review
- Risk-aware Trajectory Planning
- Iterative Risk Allocation (IRA)
- Generalizing to Risk-aware Systems
- 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%*

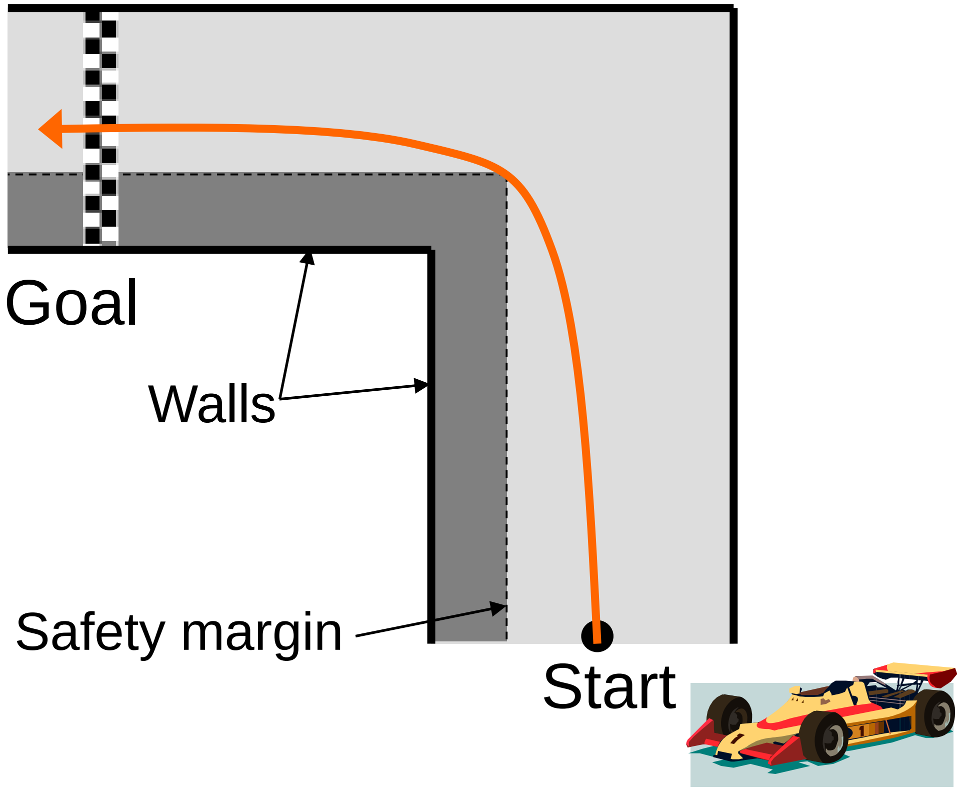
Risk bound

Idea:

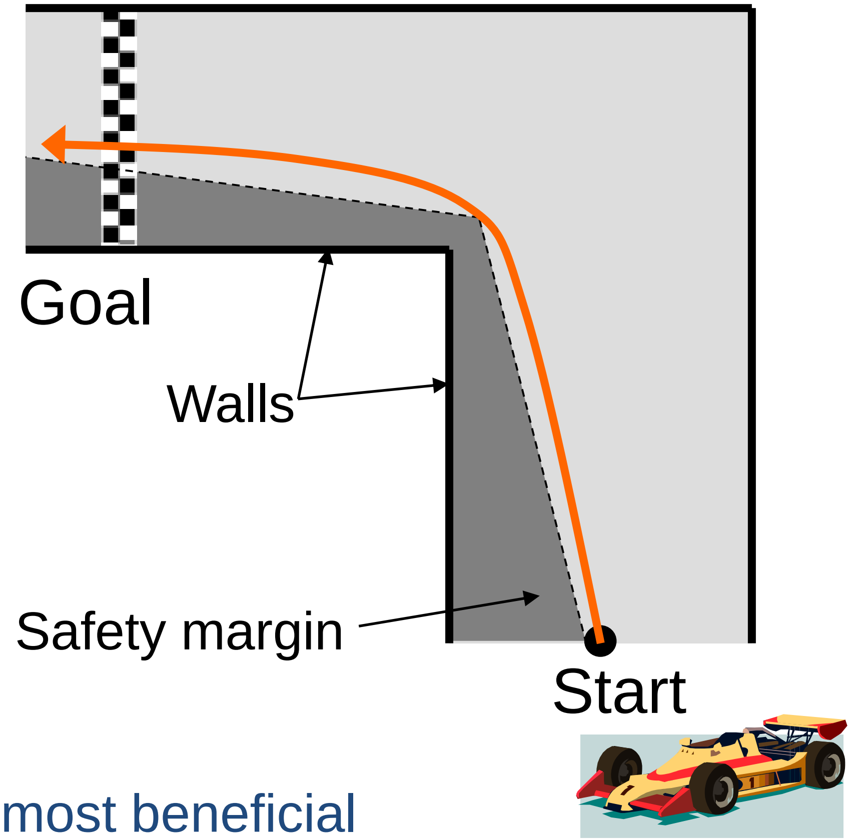
- Create safety margin that satisfies the risk bound from start to the goal.
- Reduce to simpler, deterministic optimization problem.

Executive creates safety margins that satisfy risk bounds while maximizing expected utility

(a) Uniform width safety margin



(b) Uneven width safety margin



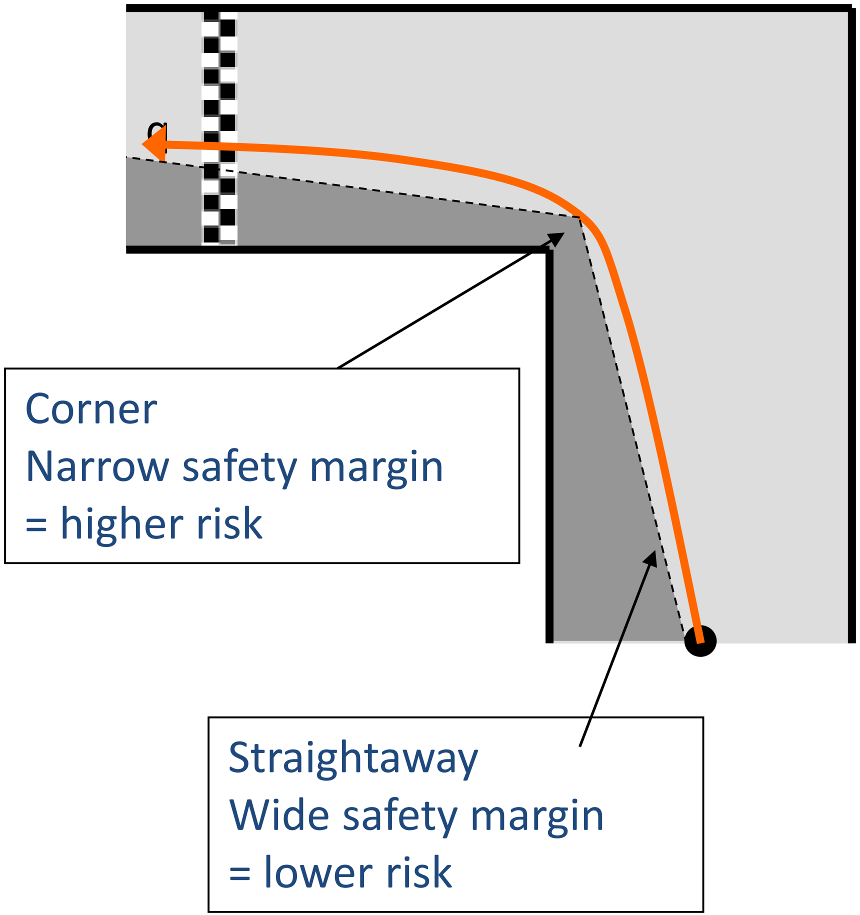
(b) results in better path → takes risk when most beneficial

Approach: Algorithmic Risk Allocation

[Ono & Williams, AAAI 08]

Key Idea - *Risk Allocation*

- Taking 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.

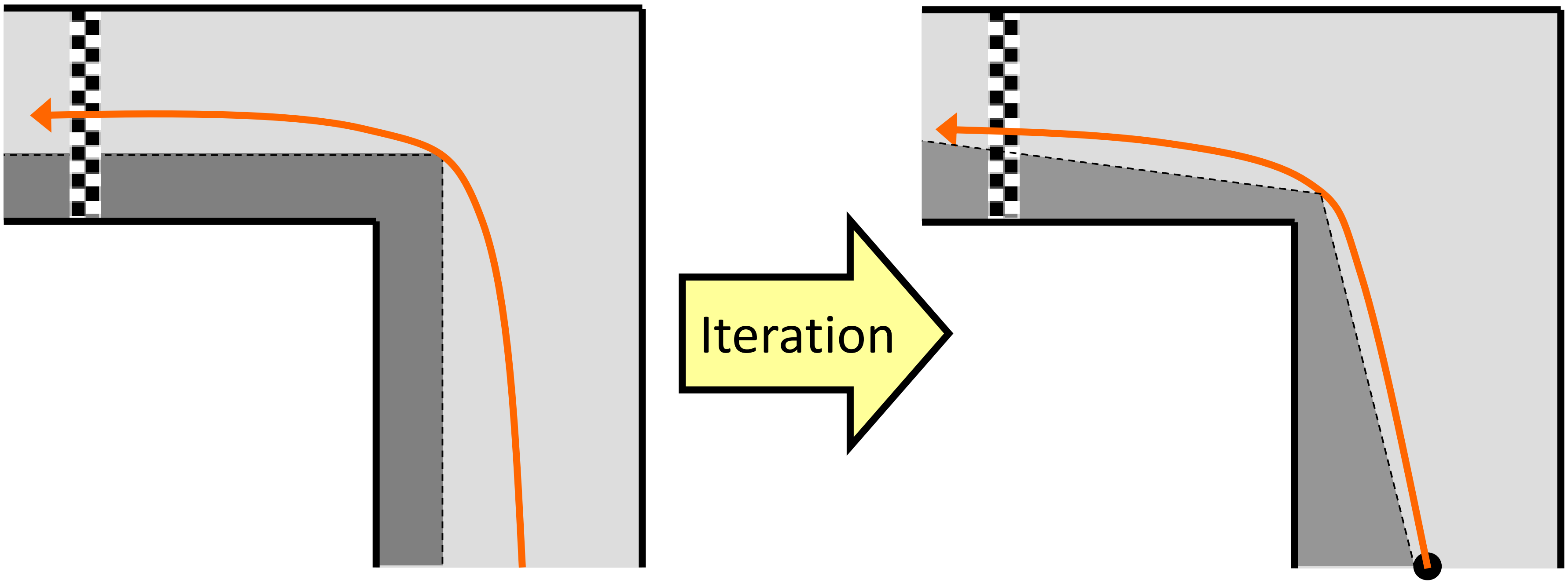


Iterative Risk Allocation (IRA) Algorithm

- Descent algorithm

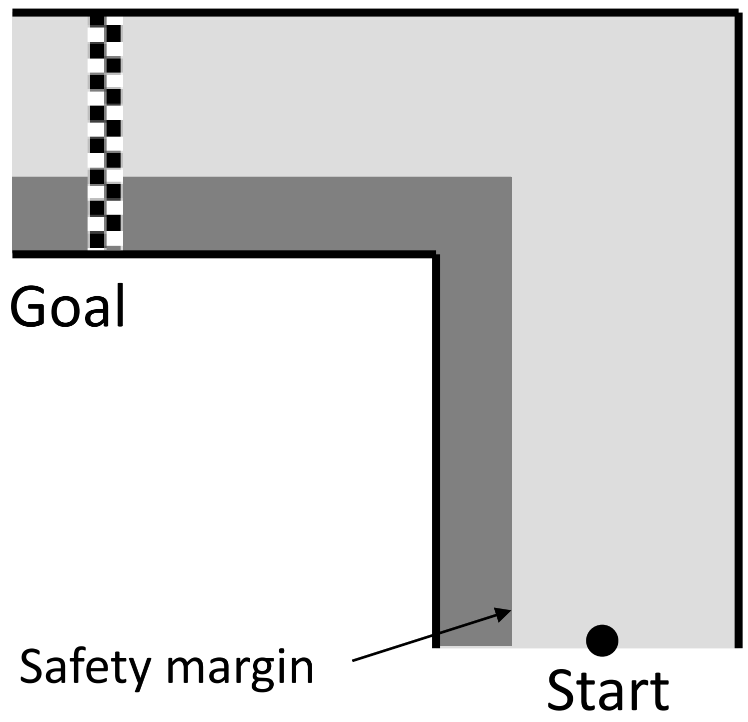
$$\bar{J}^*(\boldsymbol{\delta}_0) \geq \bar{J}^*(\boldsymbol{\delta}_1) \geq \bar{J}^*(\boldsymbol{\delta}_2) \cdots$$

(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

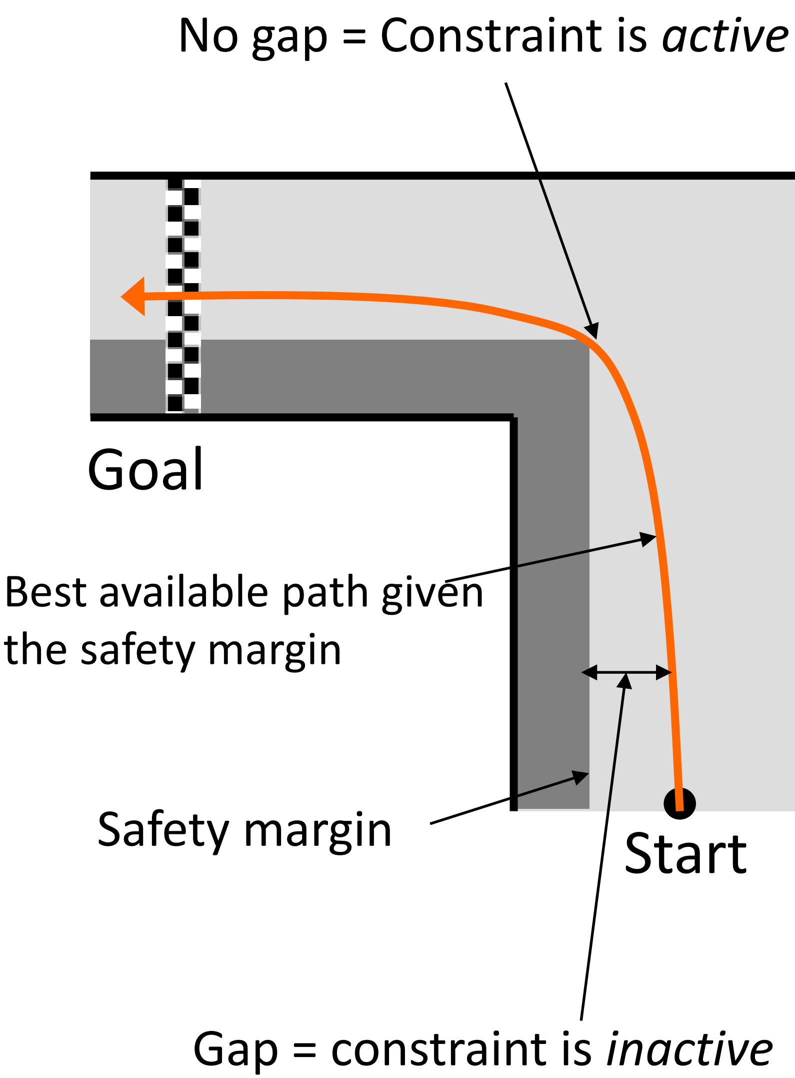
3 Compute the best available path given the current risk allocation

4 Decrease the risk where the constraint is inactive

5 Increase the risk where the constraint is active

6 End loop

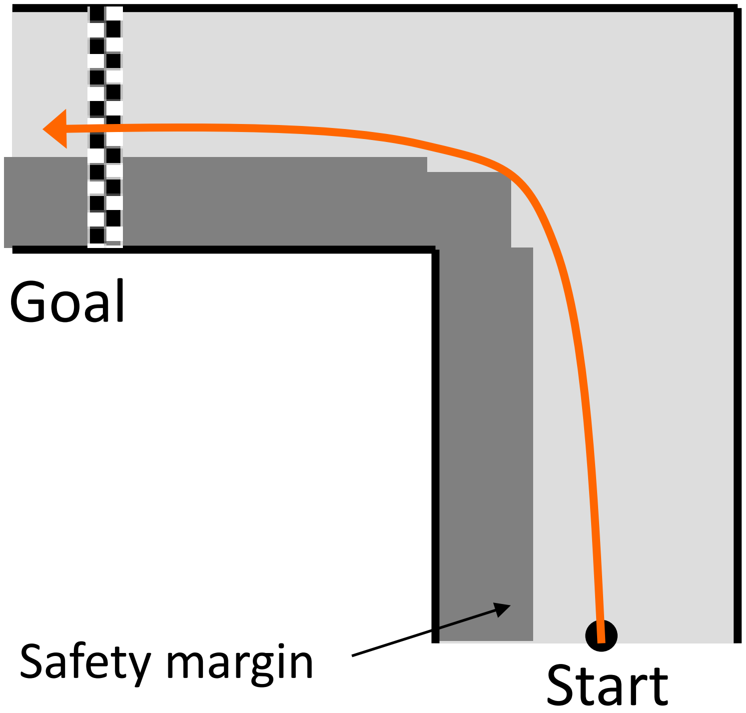
Iterative Risk Allocation Algorithm



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

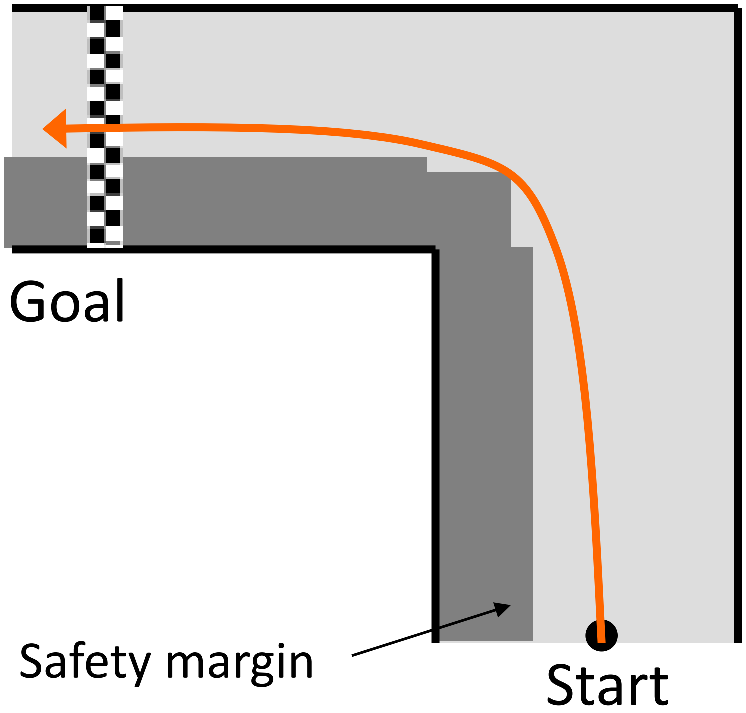
Iterative Risk Allocation Algorithm



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

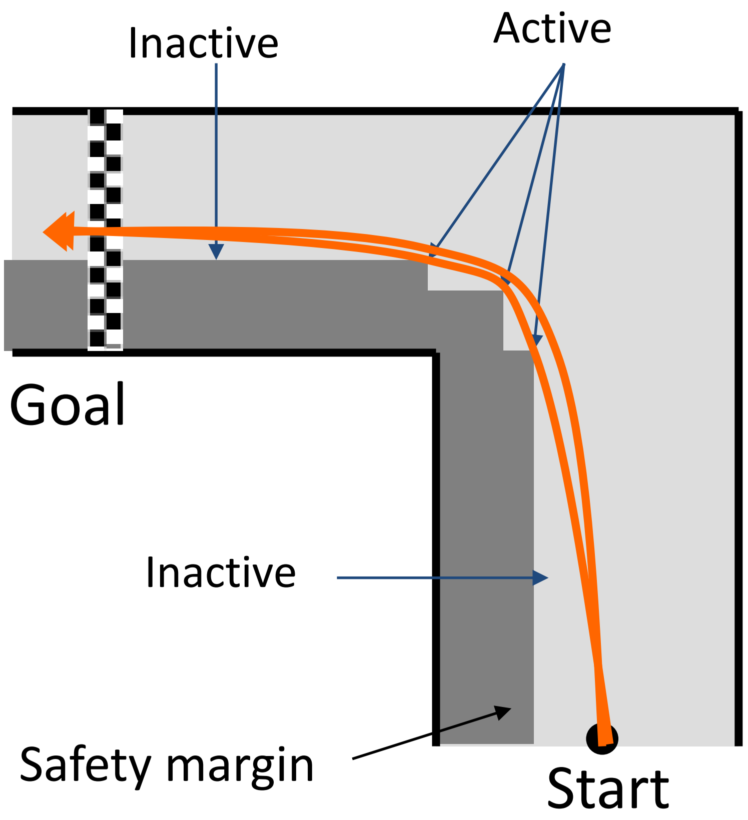
Iterative Risk Allocation Algorithm



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
5 Increase the risk where the constraint is active
6 End loop

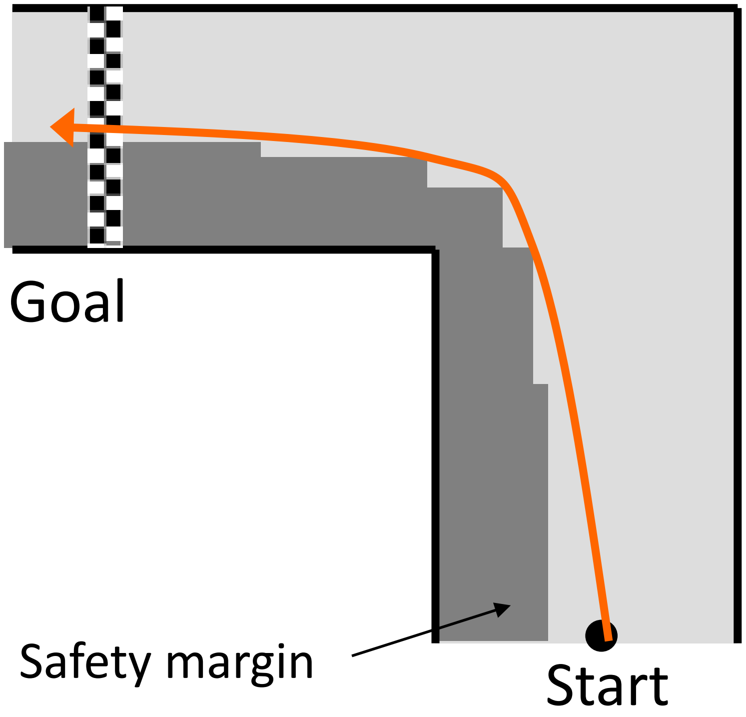
Iterative Risk Allocation Algorithm



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

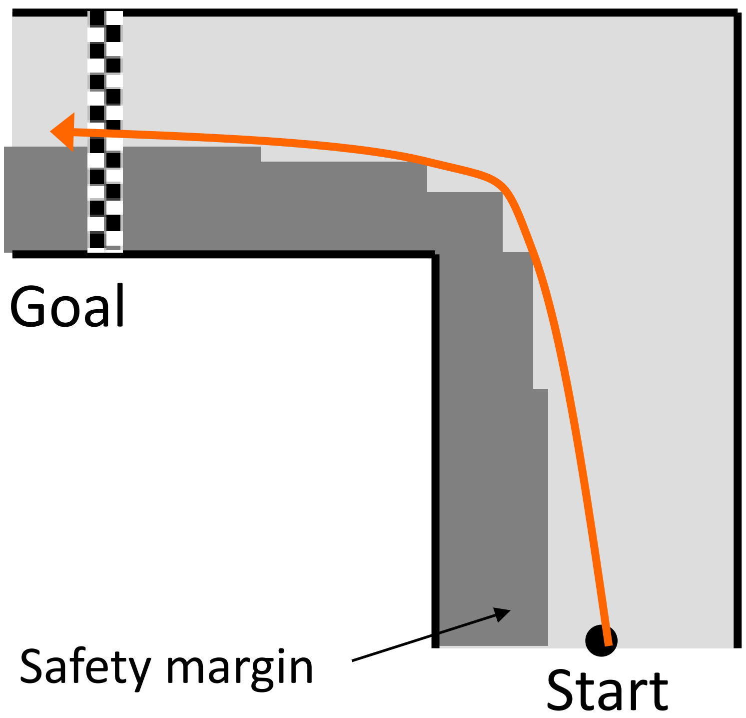
Iterative Risk Allocation Algorithm



Algorithm IRA

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

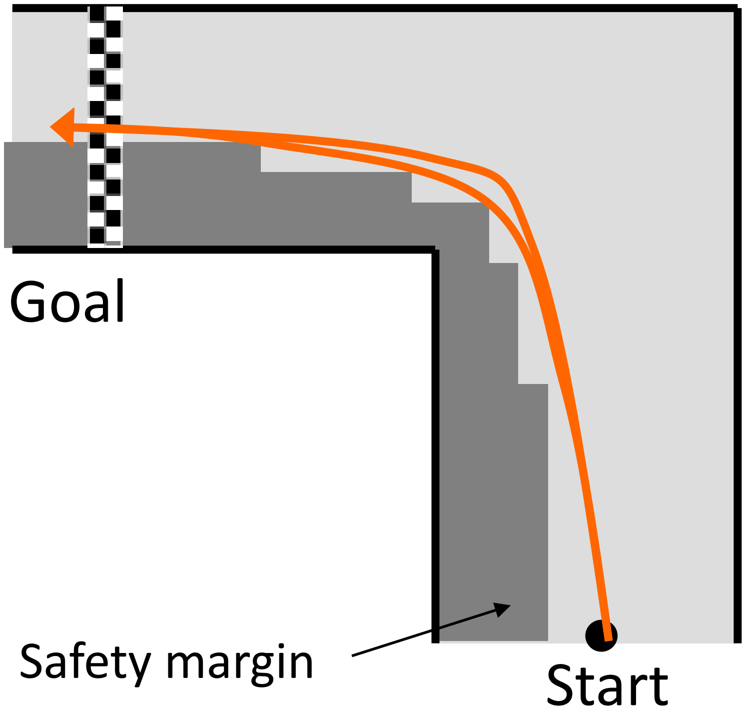
Iterative Risk Allocation Algorithm



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

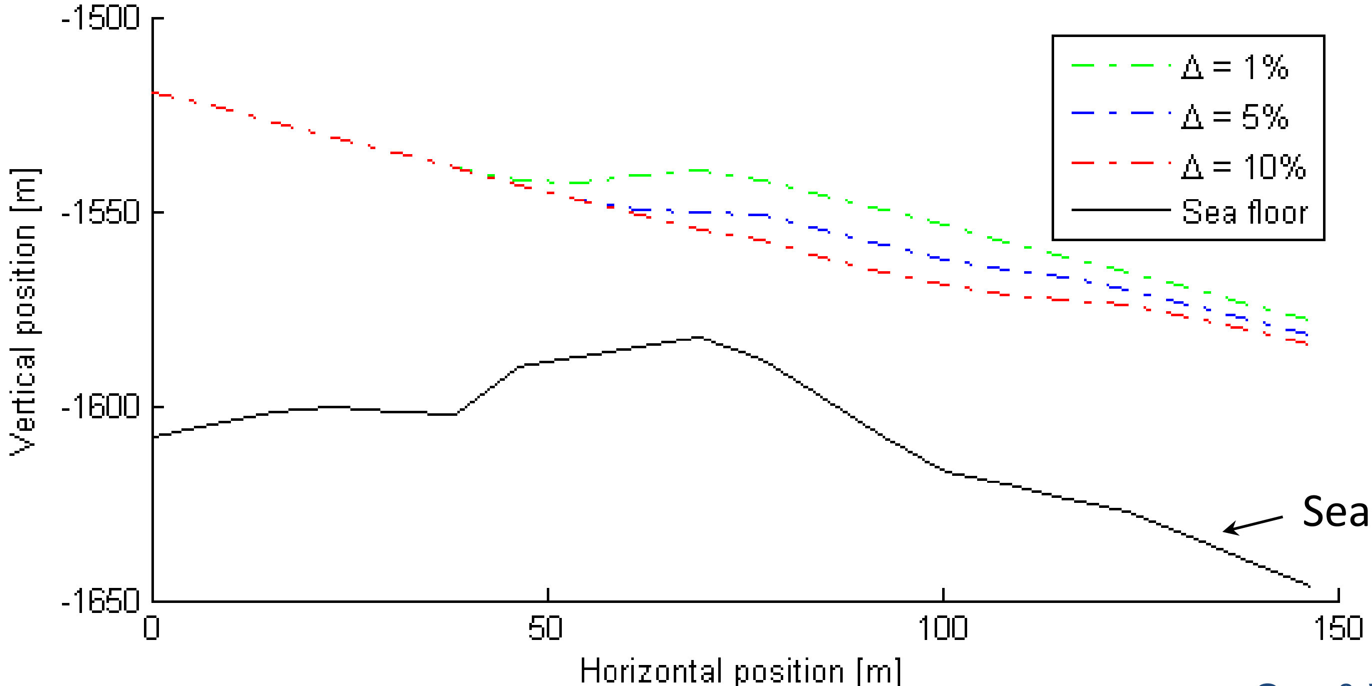
Iterative Risk Allocation Algorithm



Algorithm IRA

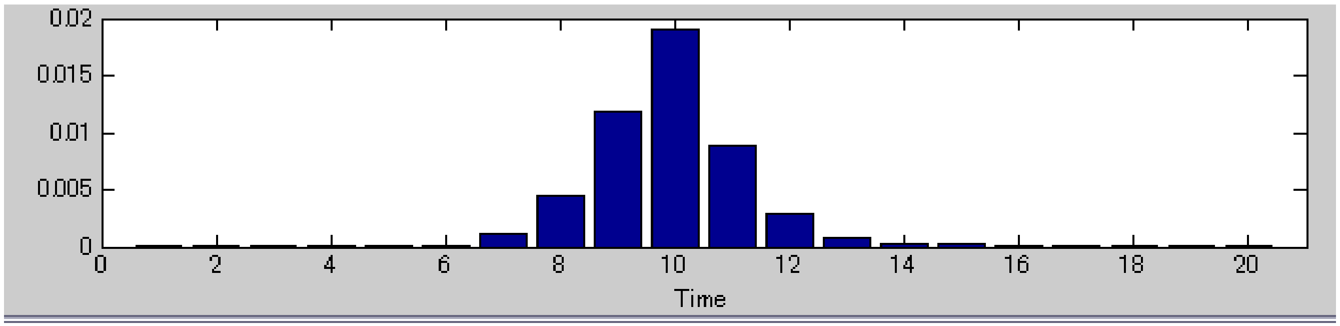
1 Initialize with arbitrary risk allocation
2 Loop
3 Compute the best available path given the current risk allocation
4 Decrease the risk where the constraint is inactive
5 Increase the risk where the constraint is active
6 End loop

Monterey Bay Mapping Example



Ono & Williams, AAAI 08

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



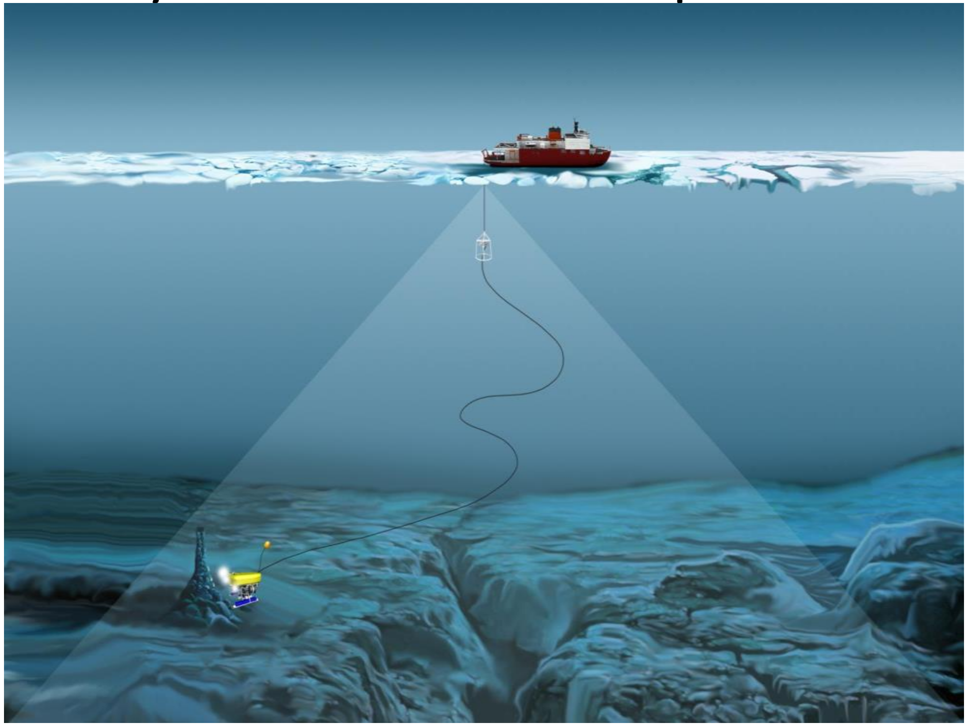
Outline

- Review
- Risk-aware Trajectory Planning (IRA)
- Iterative Risk Allocation
- Generalizing to Risk-aware Systems
- Convex Risk Allocation (CRA)

Risk-Sensitive Architectures for Exploration

- In collaboration with JPL, WHOI and Caltech.
- Initial year study, funded by Keck Institute for Spaces Sciences.

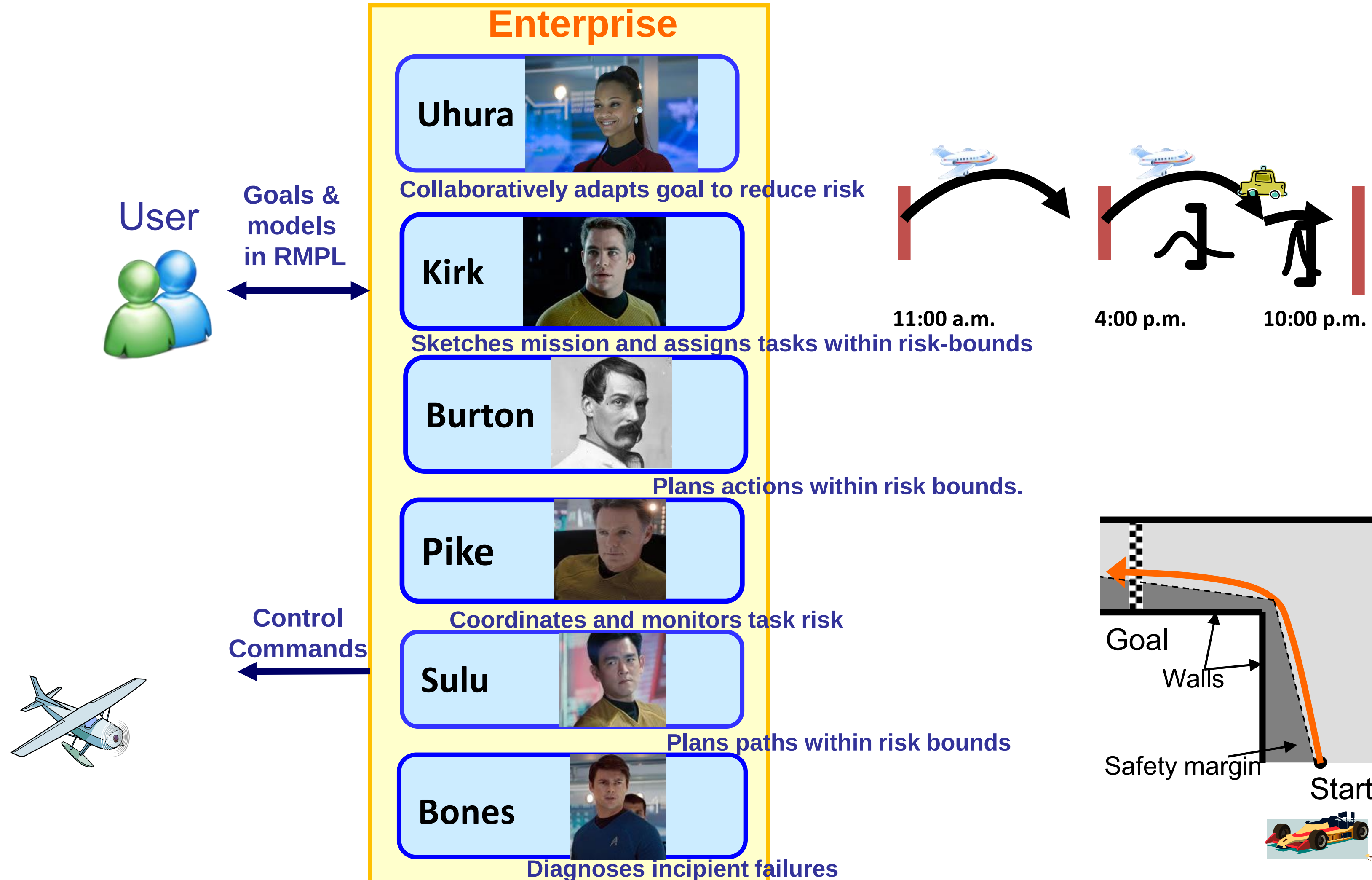
Venus Sage



NereId Under Ice

- 2 year follow on for demonstration.

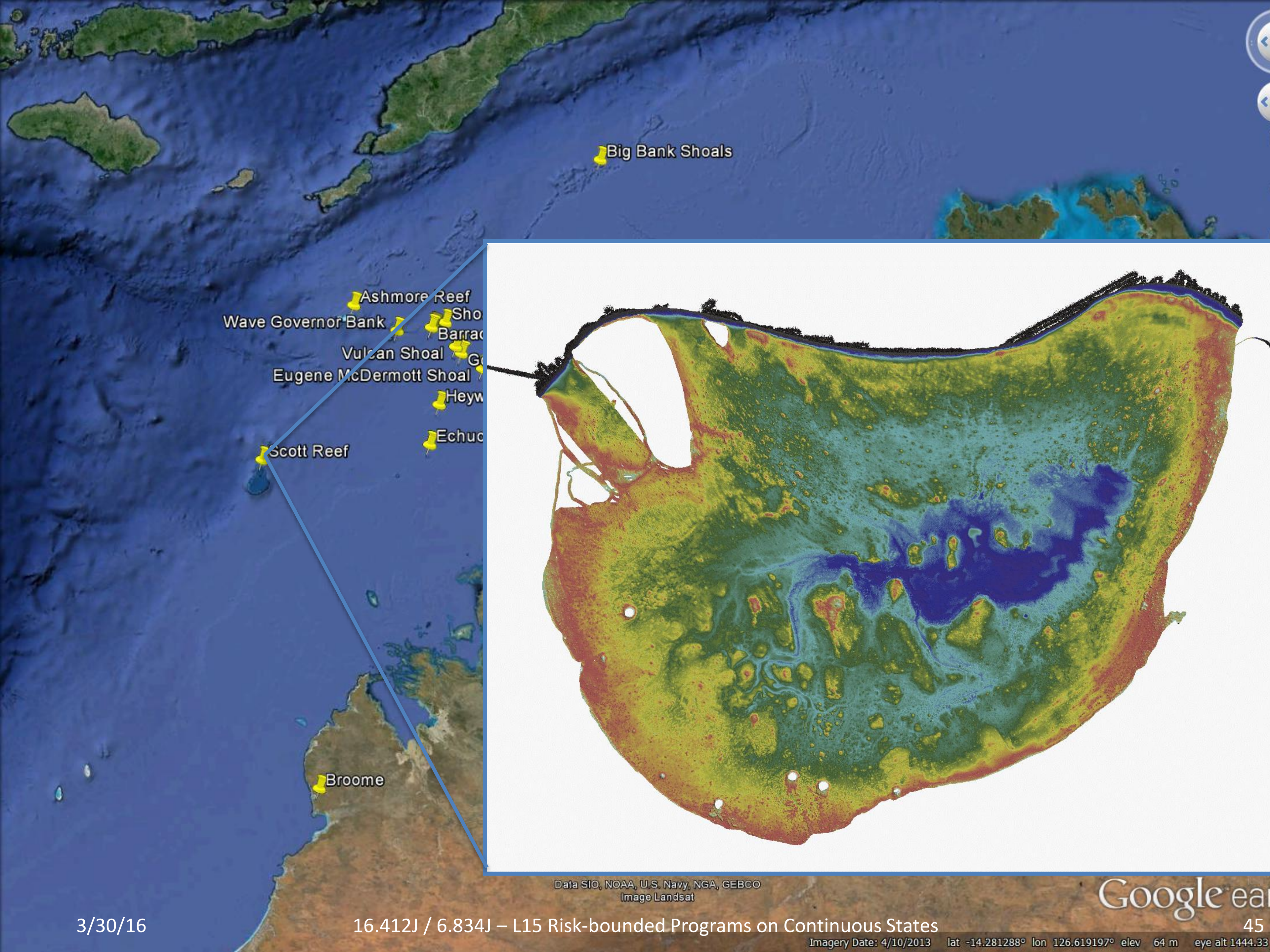
Risk-aware Planning & Execution



Falkor Cruise – March-April, 2015

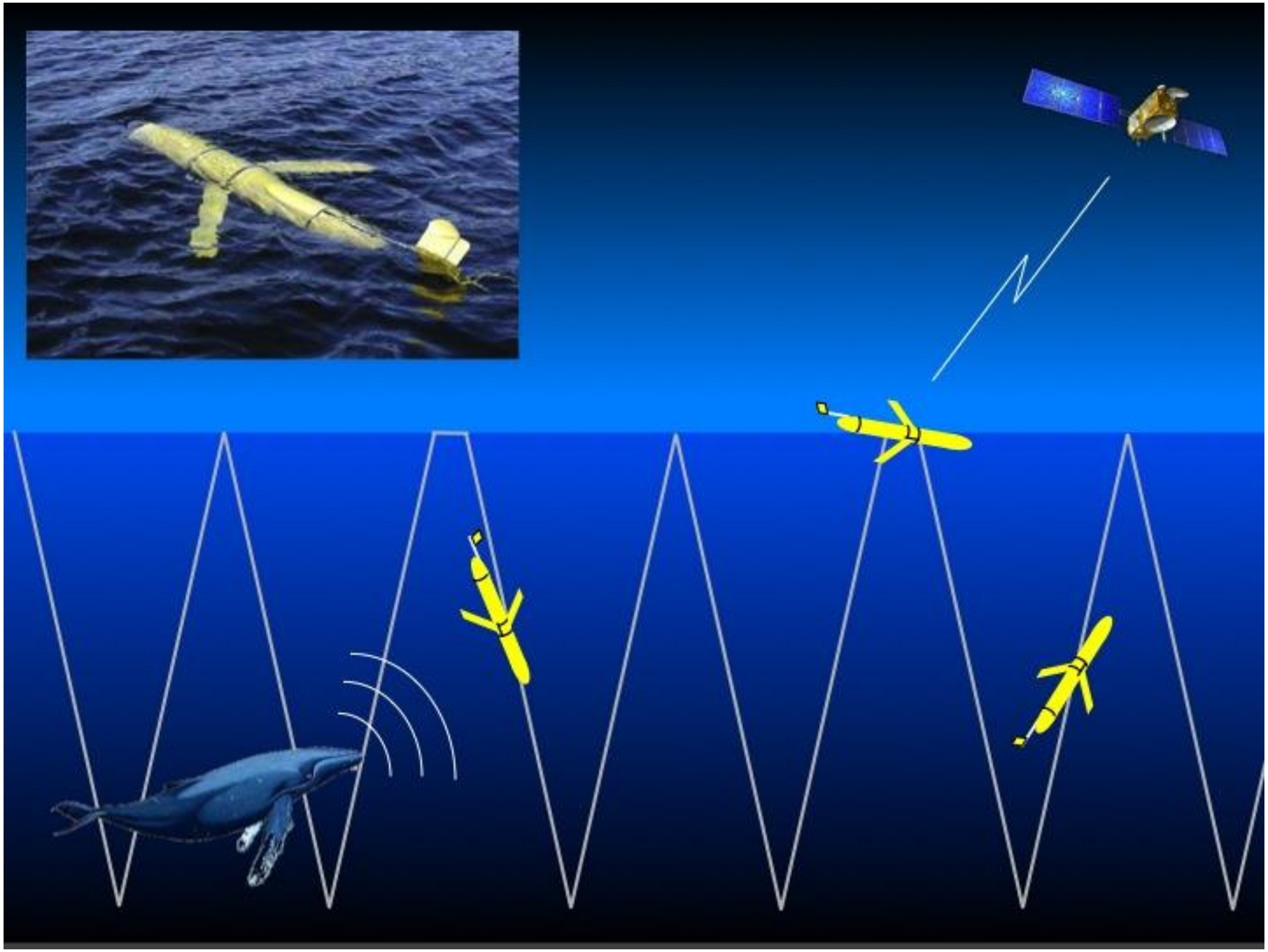
Falkor - Schmidt Ocean Institute

Slocum Glider



Big Bank Shoals
Ashmore Reef
Wave Governor Bank
Vulcan Shoal
Eugene McDermott Shoal
Scott Reef
Broome
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
Image Landsat
Imagery Date: 4/10/2013 lat -14.281288° lon 126.619197° elev 64 m

Glider Primer



User

Goals & models in RMPL

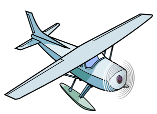
Enterprise

Kirk

Sketches mission and assigns tasks

Pike

Coordinates and monitors tasks

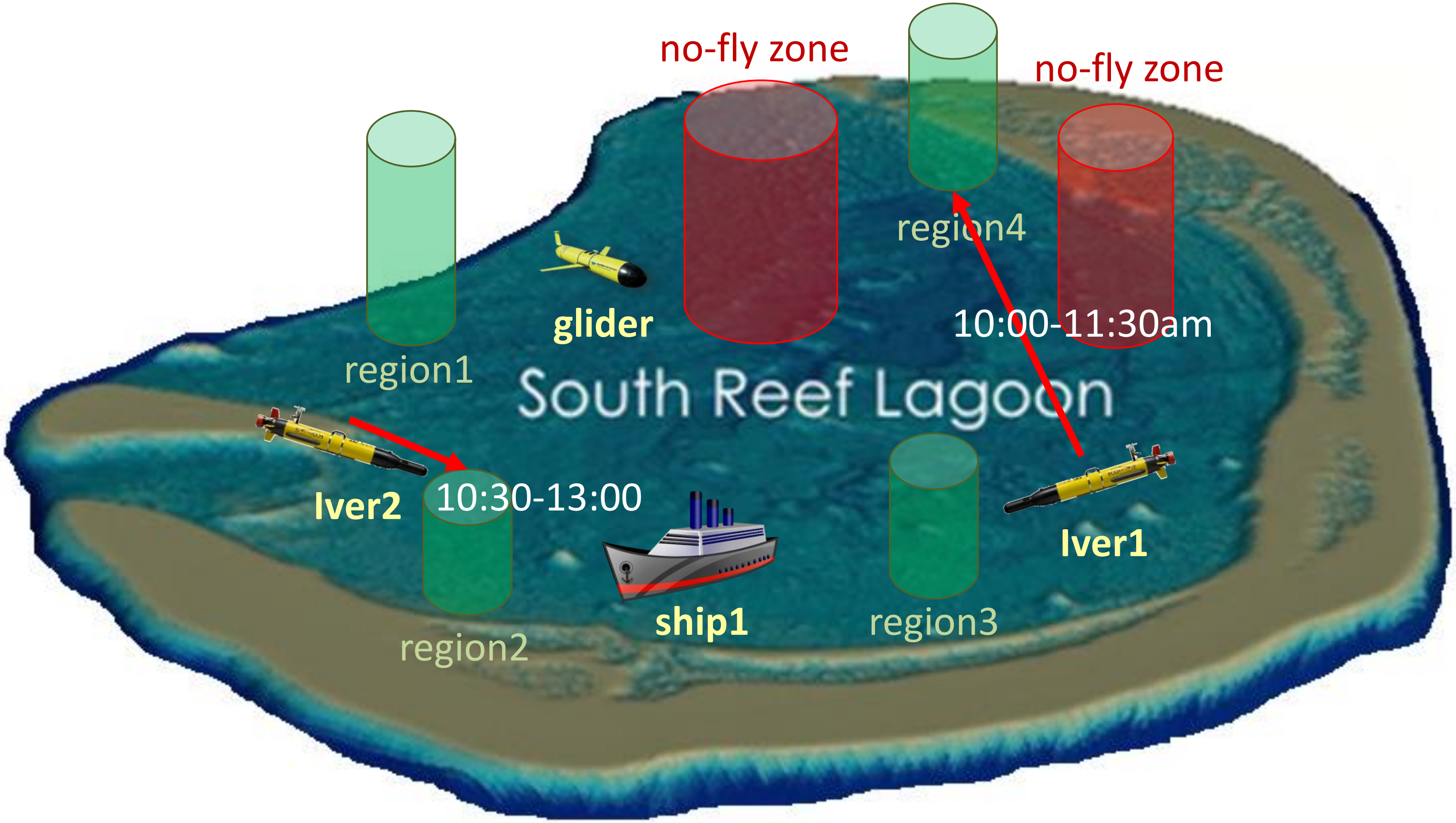


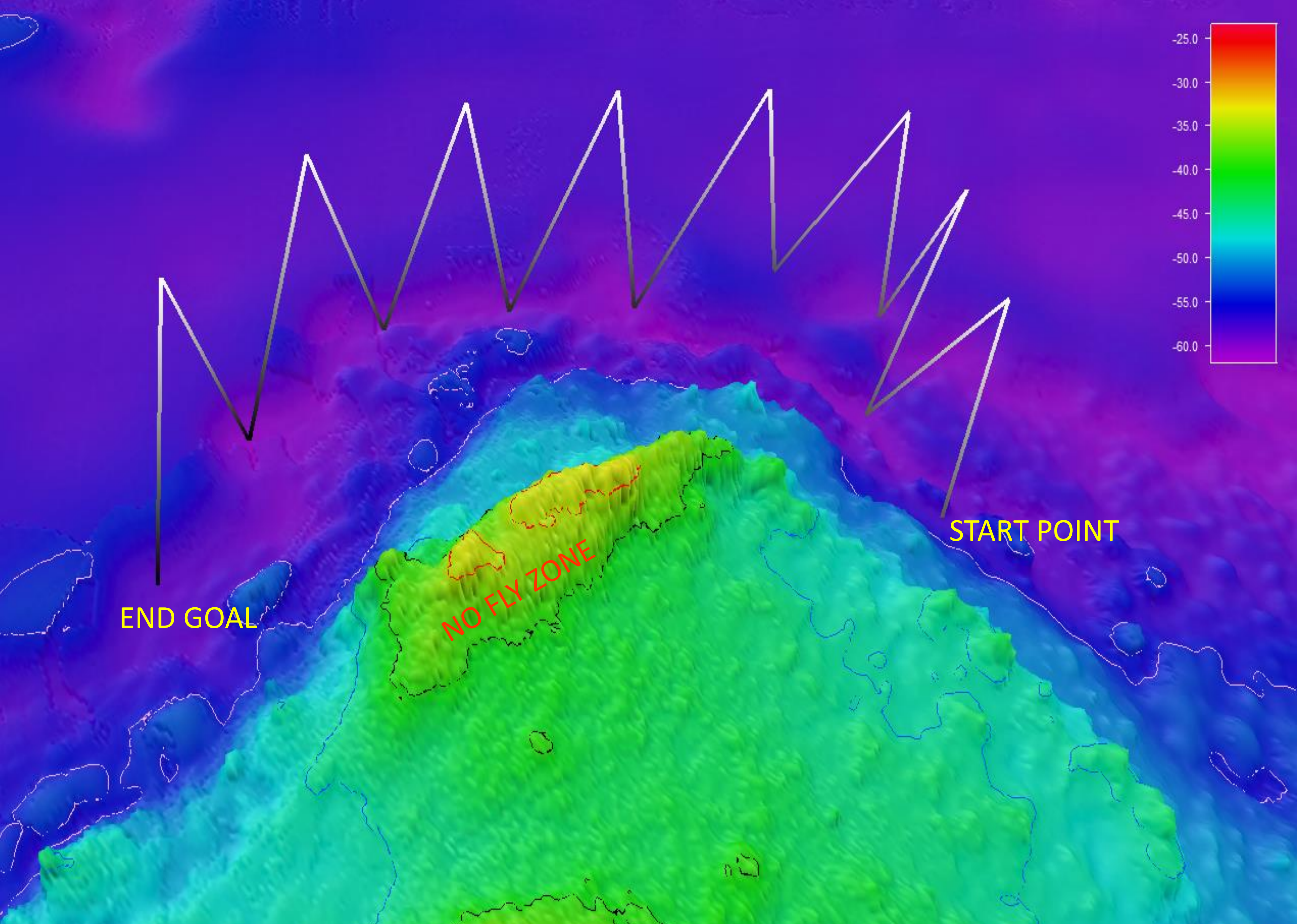
Control Commands

Sulu-lite

Plans paths

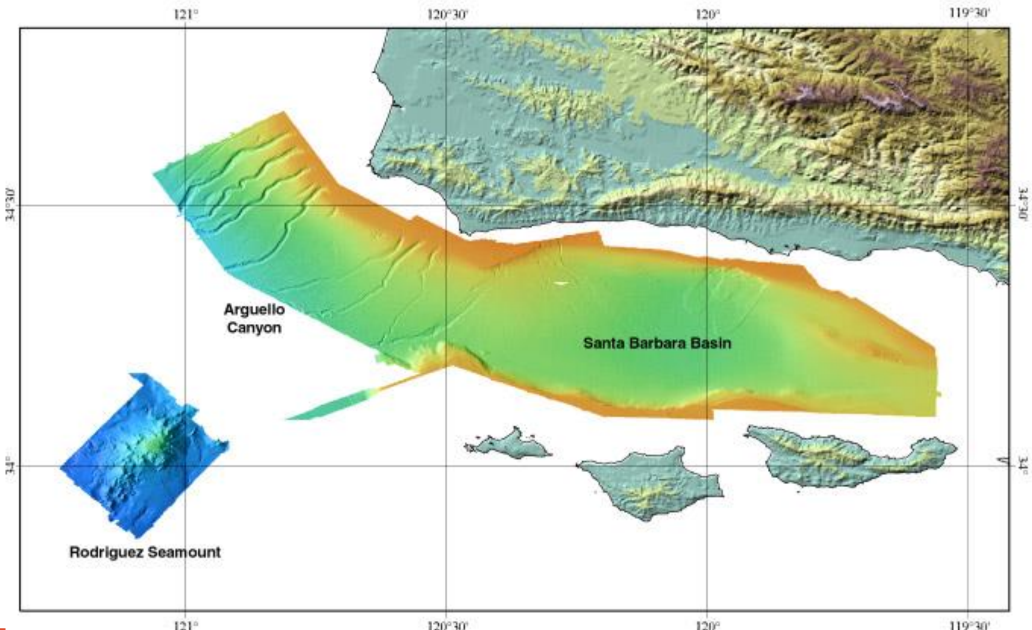
Activity Planning in Scott Reef




-25.0
-30.0
-35.0
-40.0
-45.0
-50.0
-55.0
-60.0
START POINT
END GOAL
NO FLY ZONE

April/May 2016 Deployment: Slocum Glider at Santa Barbara

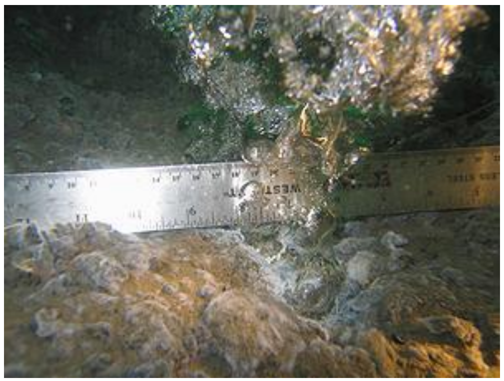
- Mission goal:
  - Use miniaturized mass spectrometer to find and characterize oil seeps off the coast of Santa Barbara.
- Primary research goal:
  - Adaptive science using planning and rule-based algorithms.
- Secondary research goal:
  - Risk-aware path planning.

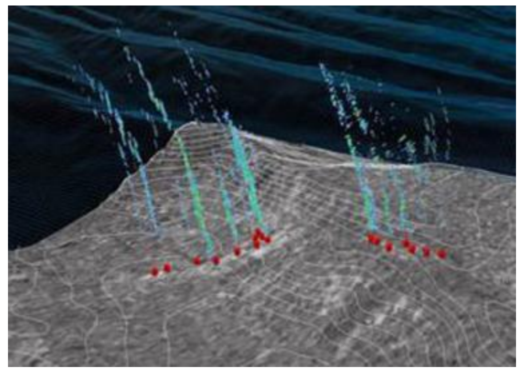


Advisory System for WHOI Cruise AT26-06: San Francisco, California to Los Angeles

Yu, Fang, Williams, Camilli, Fall 13

- Find and sample methane seeps near the coast.
- Locate and sample a 60 year-old DDT dumping site.
- Recover and replace incubators on the seafloor.

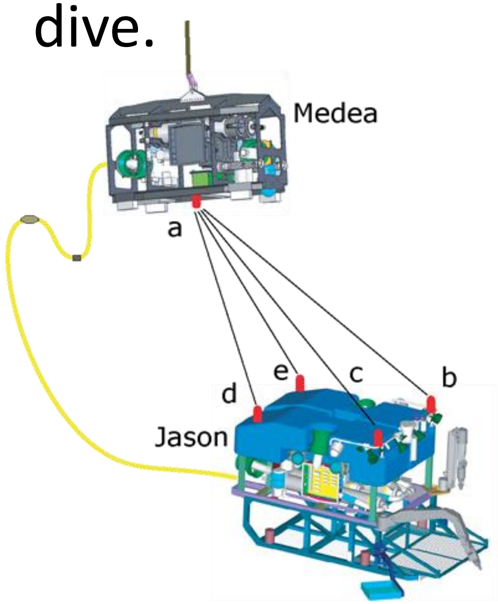




Courtesy WHOI

Everything Can Go Wrong

- [Day 1] Jason failed after 30 min into its first dive, entered an uncontrollable spin and broke its optic fiber tether.
- [Day 1] The new camera installed on Sentry did not work well in low light situations. It had been replaced during its second dive.



User

Goals in Natural Language

pEnterprise

pUhura

Collaboratively diagnoses and resolves goal failures

pKirk

Plans mission and contingencies.

PTS

Control Commands

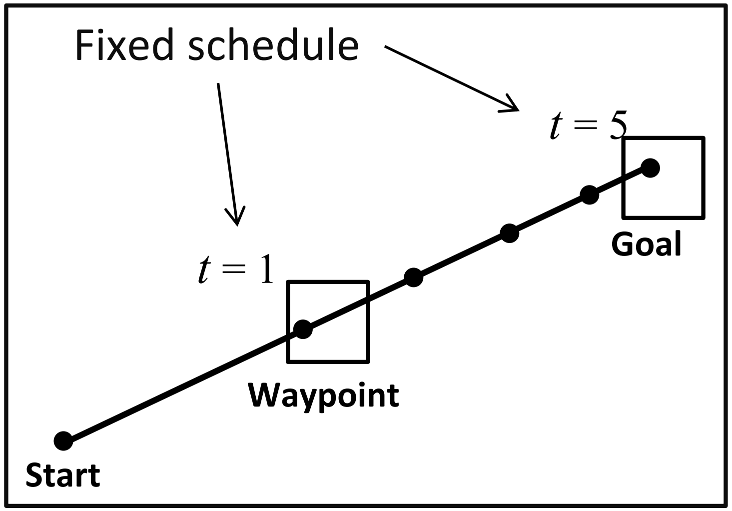
pSulu

Navigates within risk bounds

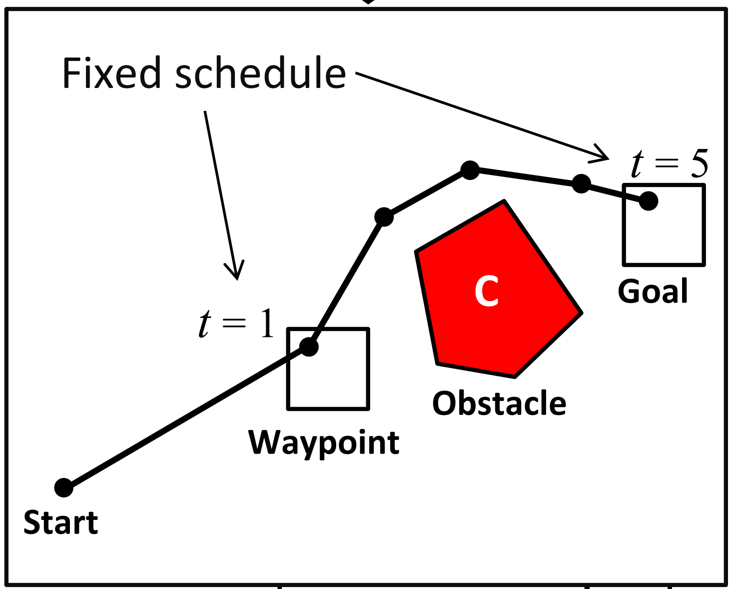
Outline

- Review
- Risk-aware Trajectory Planning
- Iterative Risk Allocation (IRA)
- Generalizing to Risk-aware Systems
- Convex Risk Allocation (CRA)
  - Intuitions
  - Math (optional)

Problems



Convex chance-constrained traj opt

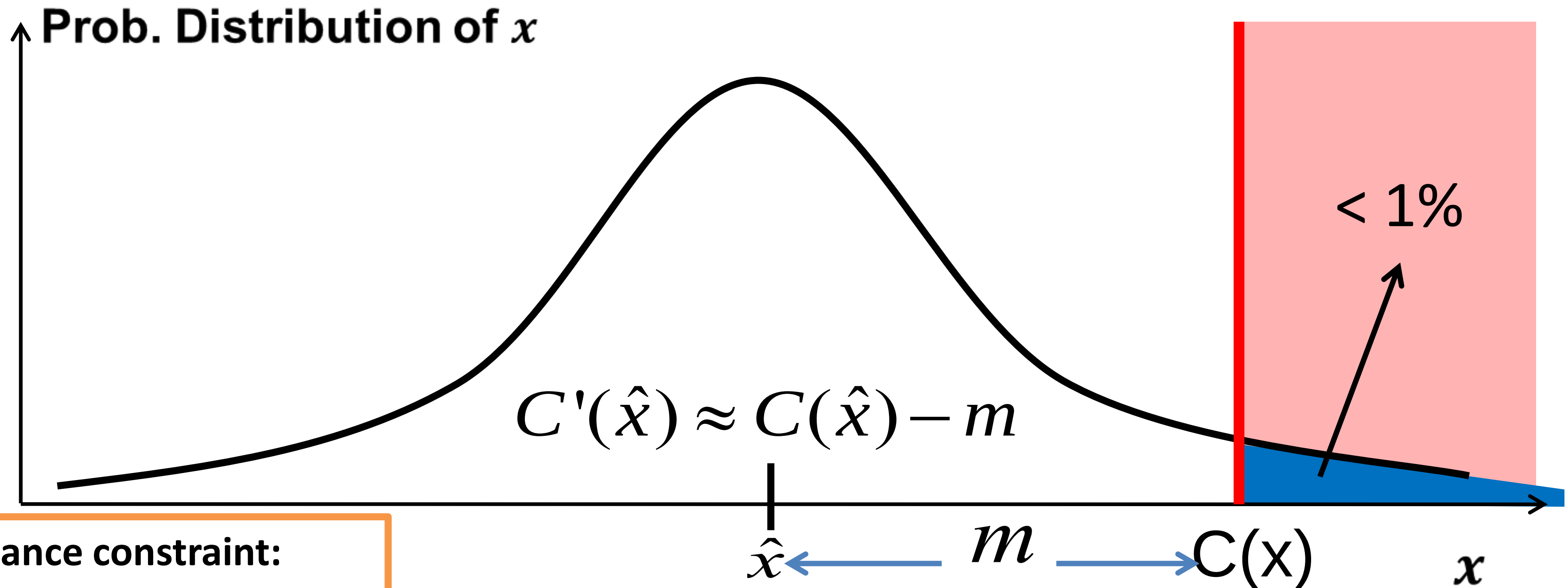


Non-convex, chance-constrained traj opt

Flexible schedule (QSP)

Risk-Allocation Overview: One Variable

Idea 1: We easily solve a chance constrained problem with one linear constraint C and one normally distributed random variable x, by reformulating C to a deterministic constraint C' on $\hat{x}$

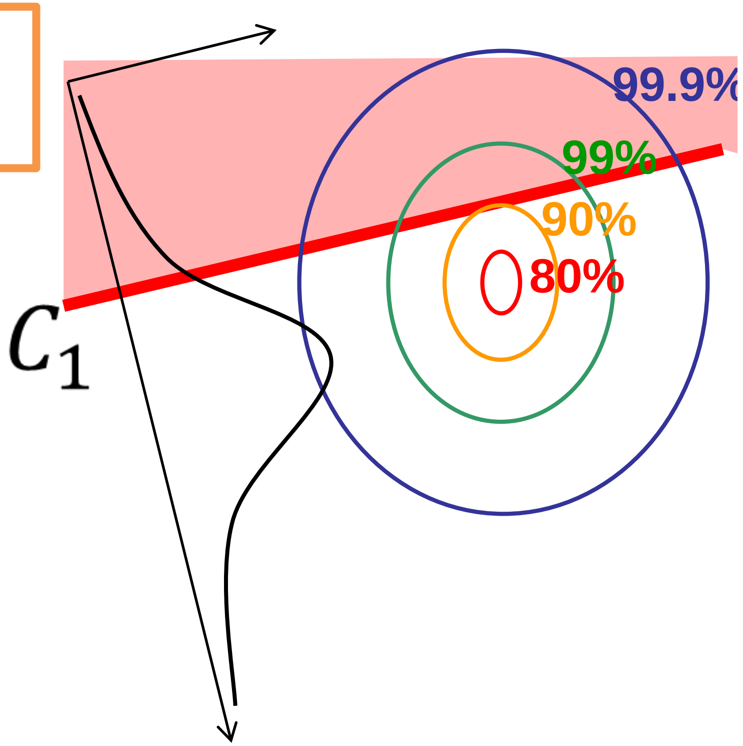


Chance constraint:
Risk < 1%

Risk-Allocation Overview: Many Variables

Chance constraint:

Risk < 1%

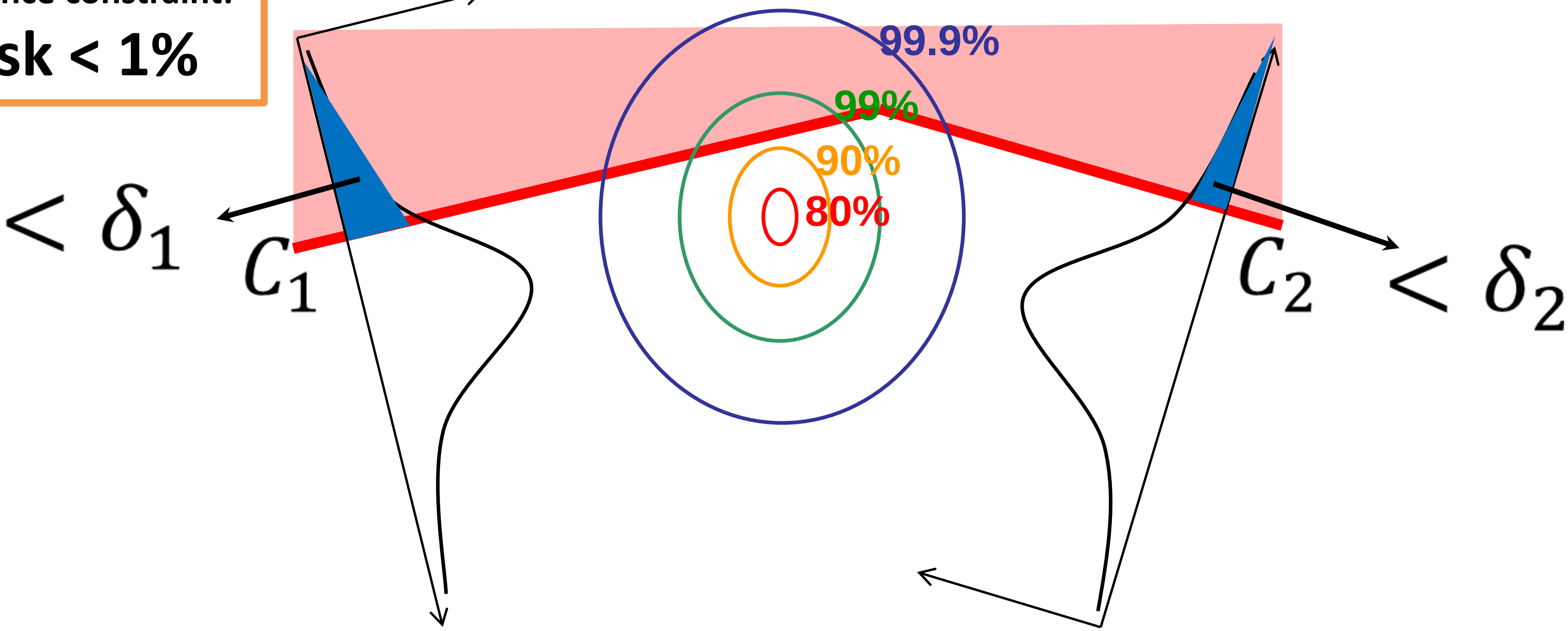


Idea 2: Generalize to a single constraint over an N-dimensional random variable, by projecting its distribution onto the axis perpendicular to the constraint boundary.

Risk-Allocation Overview: Many Constraints

Chance constraint:

Risk < 1%

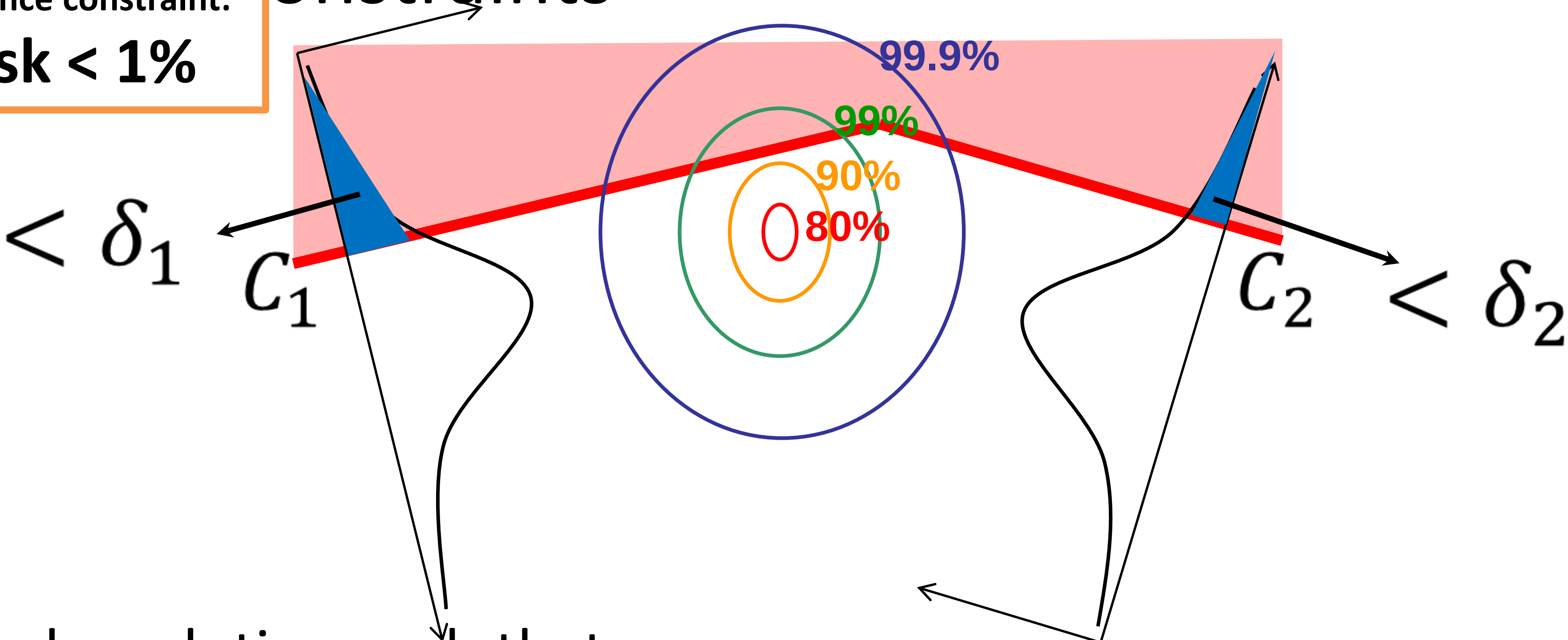


Idea 3: Generalize to a joint chance-constraint over multiple constraints C_1, C2, by distributing risk.

Risk-Allocation Overview: Constraints

Chance constraint:

Risk < 1%



Find a solution such that:

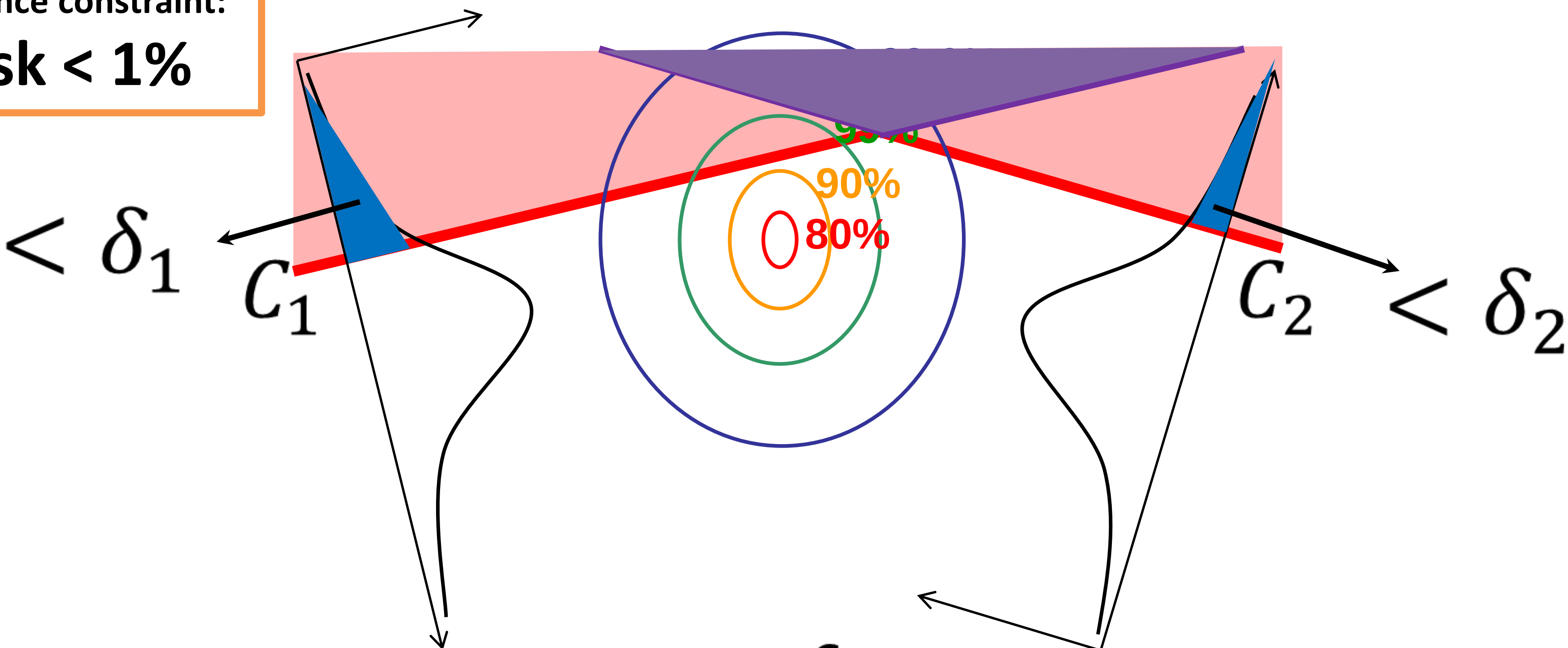
1. Each constraint C_i takes less than δ_i risk, and
2. $\Sigma_i \delta_i \leq 1\%$

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

Risk-Allocation Overview: Conservatism

Chance constraint:

Risk < 1%



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

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

Outline

- Review
- Risk-aware Trajectory Planning
- Iterative Risk Allocation (IRA)
- Generalizing to Risk-aware Systems
- Convex Risk Allocation (CRA)
  - Intuitions
  - Math (optional)

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)

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

Where A and B denote constraint failures

1 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

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

Variable

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

Upper bound of the probability of violating *i*th 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

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

Joint chance constraint

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

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

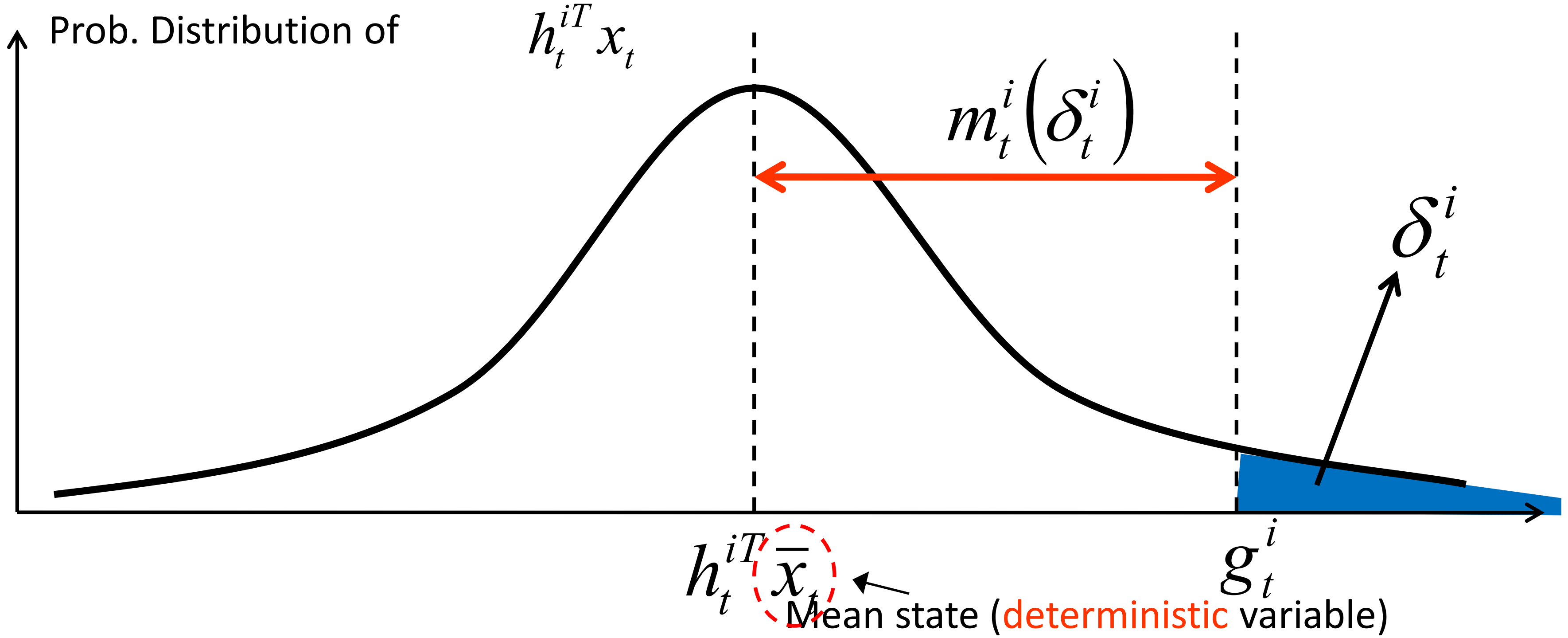
Conversion to Deterministic Constraint

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

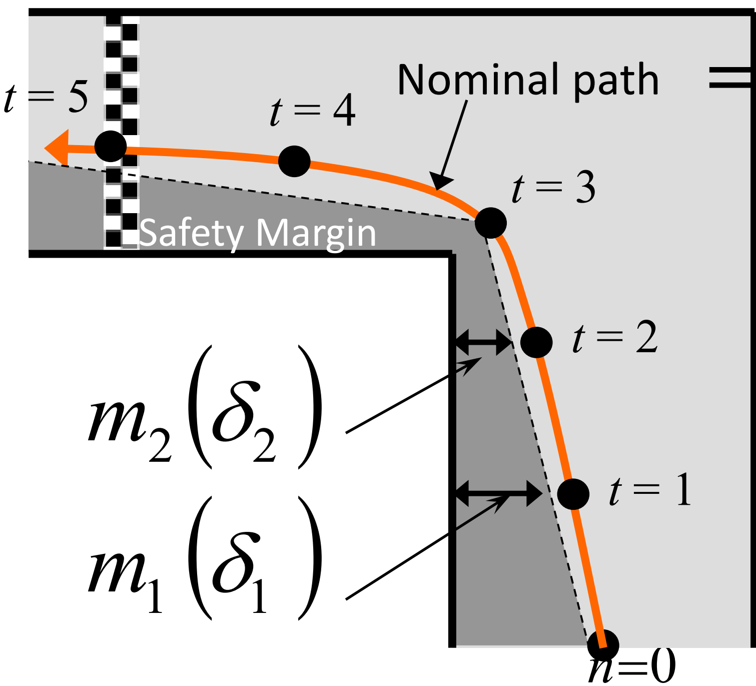
$$\Pr\left[h_t^{iT} x_t \le g_t^i\right] \ge 1 - \delta_t^i$$

Deterministic constraint

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

Mean state

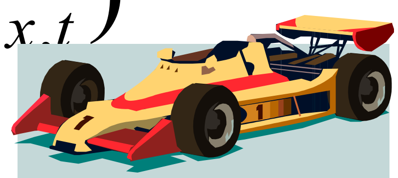
Safety Margin



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

(Inverse of cdf of Gaussian)

$$, \Sigma_{x,t})$$



$$^n \Sigma_w \qquad x,0$$

[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

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

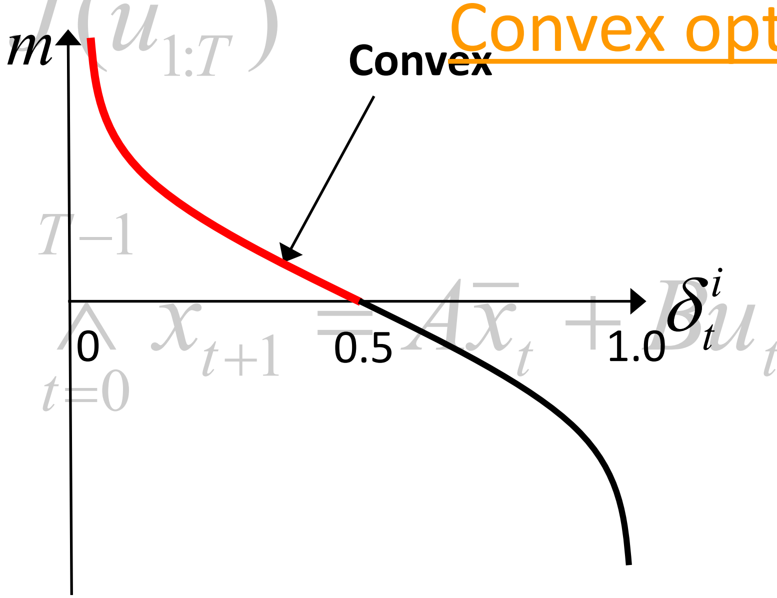
$$\sum_{t,i} \delta_t^i \le \Delta$$

Conversion to Deterministic Constraint

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

Convex optimization

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



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

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

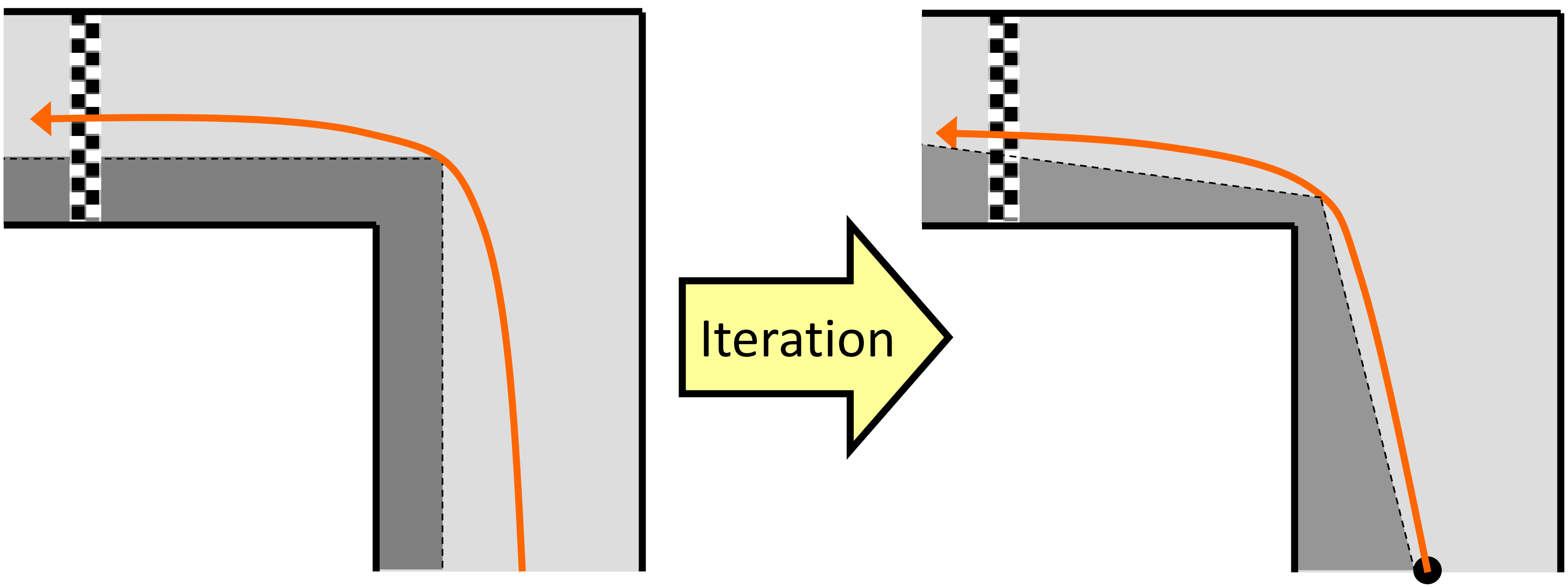
Convex if $\delta < 0.5$

Key takeaways

- Maximizing utility under bounded risk makes sense.
- Risk allocation can help us solve.

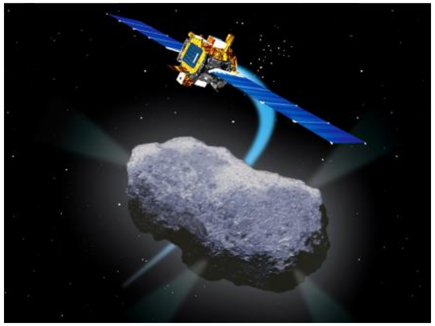
Summary: Risk Allocation

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



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

Approach: Programming Cognitive Systems

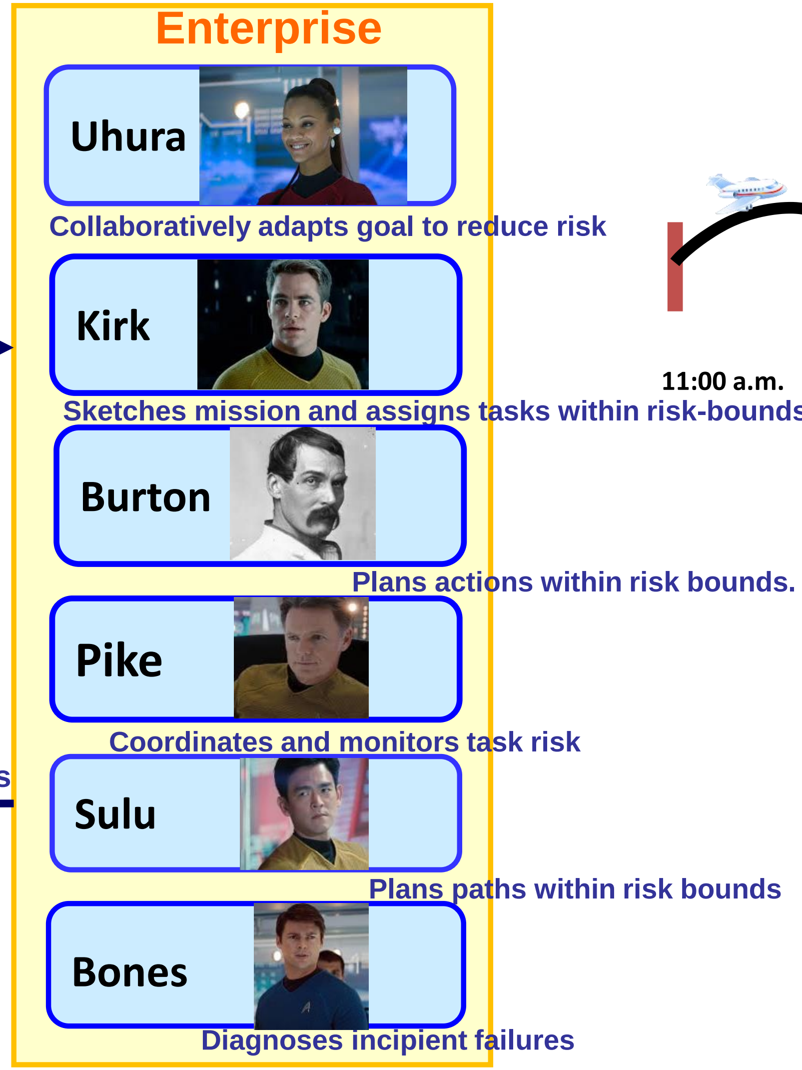


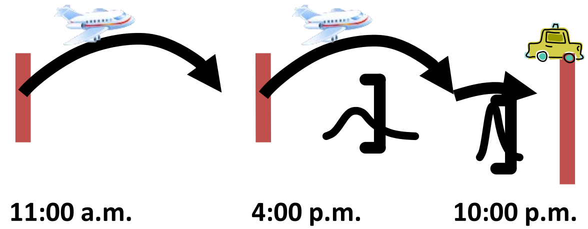
1. An embedded programming language elevated to the goal-level through partial specification and operations on state (RMPL).
2. A language executive that achieves robustness by reasoning over constraint-based models and by bounding risk (Enterprise).
3. Interfaces that support natural human interaction fluidly and at the cognitive level (Uhura).

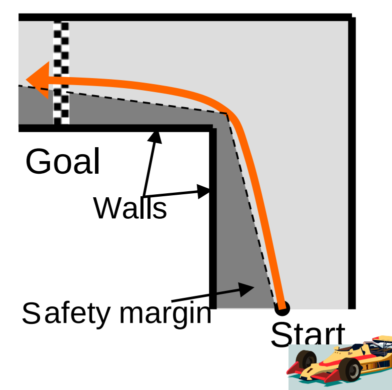
Risk_aware Planning & Execution

User

Goals & models in RMPL



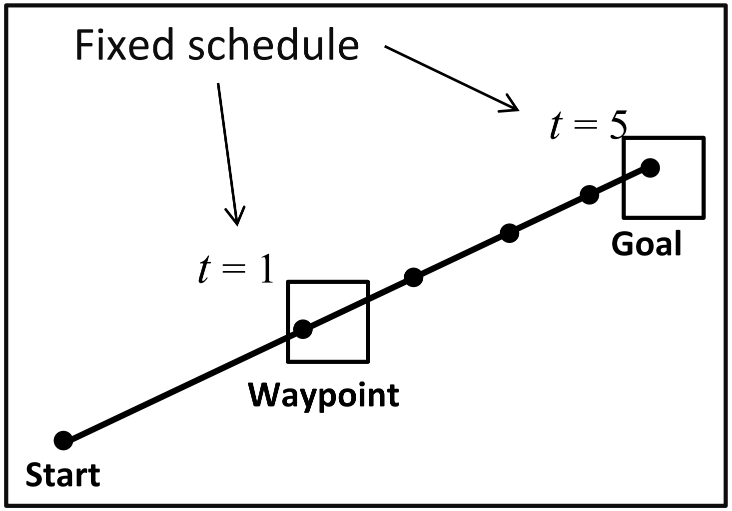




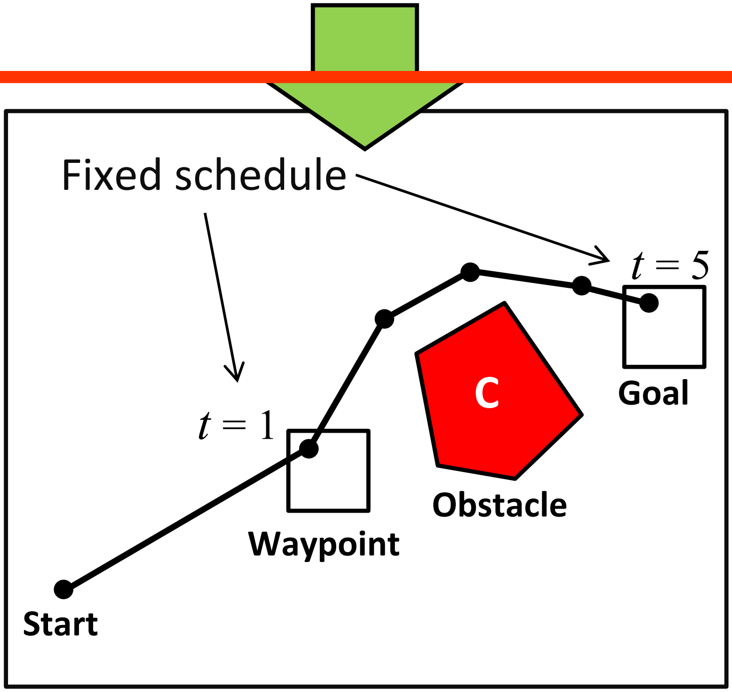
Control Commands

APPENDIX: RISK-BOUNDED PLANNING FOR NON-CONVEX PROBLEMS

Next …

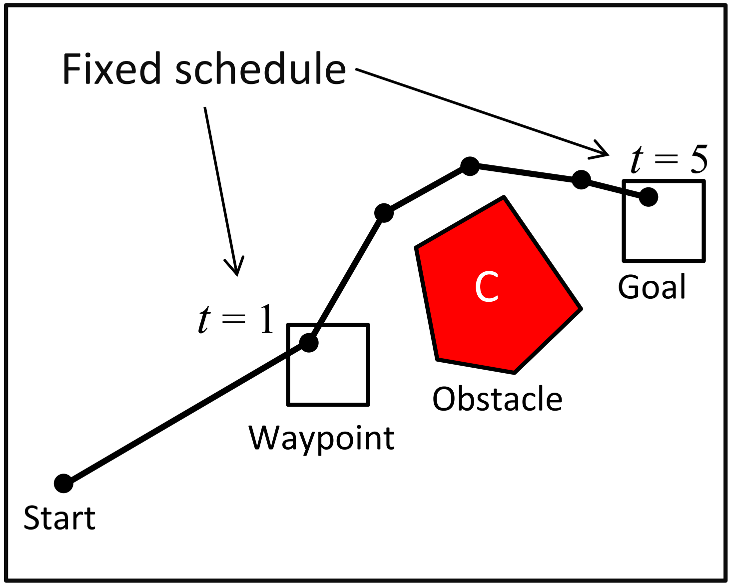


Convex chance-constrained trajectory optimization



Non-convex, chance-constrained traj opt

Non-Convex Problem Formulation



$$\min_{\boldsymbol{u}_{1:N} \in \mathbb{U}^N} \quad J'(\boldsymbol{u}_{1:N}, \bar{\boldsymbol{x}}_{1:N})$$

$$\text{s.t.} \quad \forall t \in \mathbb{T}, \quad \boldsymbol{x}_{t+1} = \boldsymbol{A}_t \boldsymbol{x}_t + \boldsymbol{B}_t \boldsymbol{u}_t + \boldsymbol{w}_t$$

$$\bigwedge_{c \in \mathcal{C}} \Pr \left[\bigwedge_{i \in \mathcal{I}_c(s)} \boxed{\bigvee_{j \in \mathcal{J}_{c,i}}} \boldsymbol{h}_{c,i,j}^T \boldsymbol{X} \leq g_{c,i,j} \right] \geq 1 - \Delta_c$$

Non-convex state constraint

Problem Formulation: Non-Convex Chance Constraint

CC over convex state constraints

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

$$s.t. \quad \Pr\left[\bigwedge_{t=1}^{T} \bigwedge_{n=1}^{N} h_t^{nT} x_t \le g_t^n \right] \ge 1 - \Delta$$

Risk allocation

Convex Deterministic Program

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

$$s.t. \quad \bigwedge_{t=1}^{T} \bigwedge_{n=1}^{N} h_t^{nT} \bar{x}_t \le g_t^n - m_t^n\left(\delta_t^n\right)$$

$$\sum_{t=1,n=1}^{T,N} \delta_t^n \le \Delta$$

A joint chance constraints

1

A set of individual chance constraints

2

A set of deterministic state constraints

Problem Formulation: Non-Convex Chance Constraint

CC over convex state constraints

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

$$s.t.\quad \Pr\left[\bigwedge_{t=1}^{T}\bigwedge_{n=1}^{N} h_t^{nT} x_t \le g_t^n\right] \ge 1-\Delta$$

Risk allocation

Convex Deterministic Program

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

$$s.t.\quad \bigwedge_{t=1}^{T}\bigwedge_{n=1}^{N} h_t^{nT}\bar{x}_t \le g_t^n - m_t^n\left(\delta_t^n\right)$$

$$\sum_{t=1,n=1}^{T,N}\delta_t^n \le \Delta$$

CC over non-convex state constraints

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

$$s.t.\quad \Pr\left[\bigwedge_{t=1}^{T}\bigwedge_{n=1}^{N}\boxed{\bigvee_{k=1}^{K}} h_t^{n,kT} x_t \le g_t^{n,k}\right] \ge 1-\Delta$$

Risk allocation

Risk selection

Convex Disjunctive Deterministic Program

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

$$s.t.\quad \bigwedge_{t=1}^{T}\bigwedge_{n=1}^{N}\boxed{\bigvee_{k=1}^{K}} h_t^{n,kT}\bar{x}_t \le g_t^{n,k} - m_t^{n,k}\left(\delta_t^n\right)$$

$$\sum_{t=1,n=1}^{T,N}\delta_t^n \le \Delta$$

Solving Disjunctive Program using Branch and Bound

Example:

$$\bigwedge_{t=1}^{T}\bigwedge_{n=1}^{N}\bigvee_{k=1}^{K} h_t^{n,kT}\bar{x}_t \le g_t^{n,k} - m_t^{n,k}\left(\delta_t^n\right)$$

$$= (C_{11} \vee C_{12}) \wedge (C_{21} \vee C_{22})$$

$$T=2, N=1, K=2; \quad C_{tk} \equiv \left\{ h_t^{1,kT}\bar{x}_t \le g_t^{1,k} - m_t^{1,k}\left(\delta_t^n\right)\right\}$$

Convex Disjunctive Programming

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

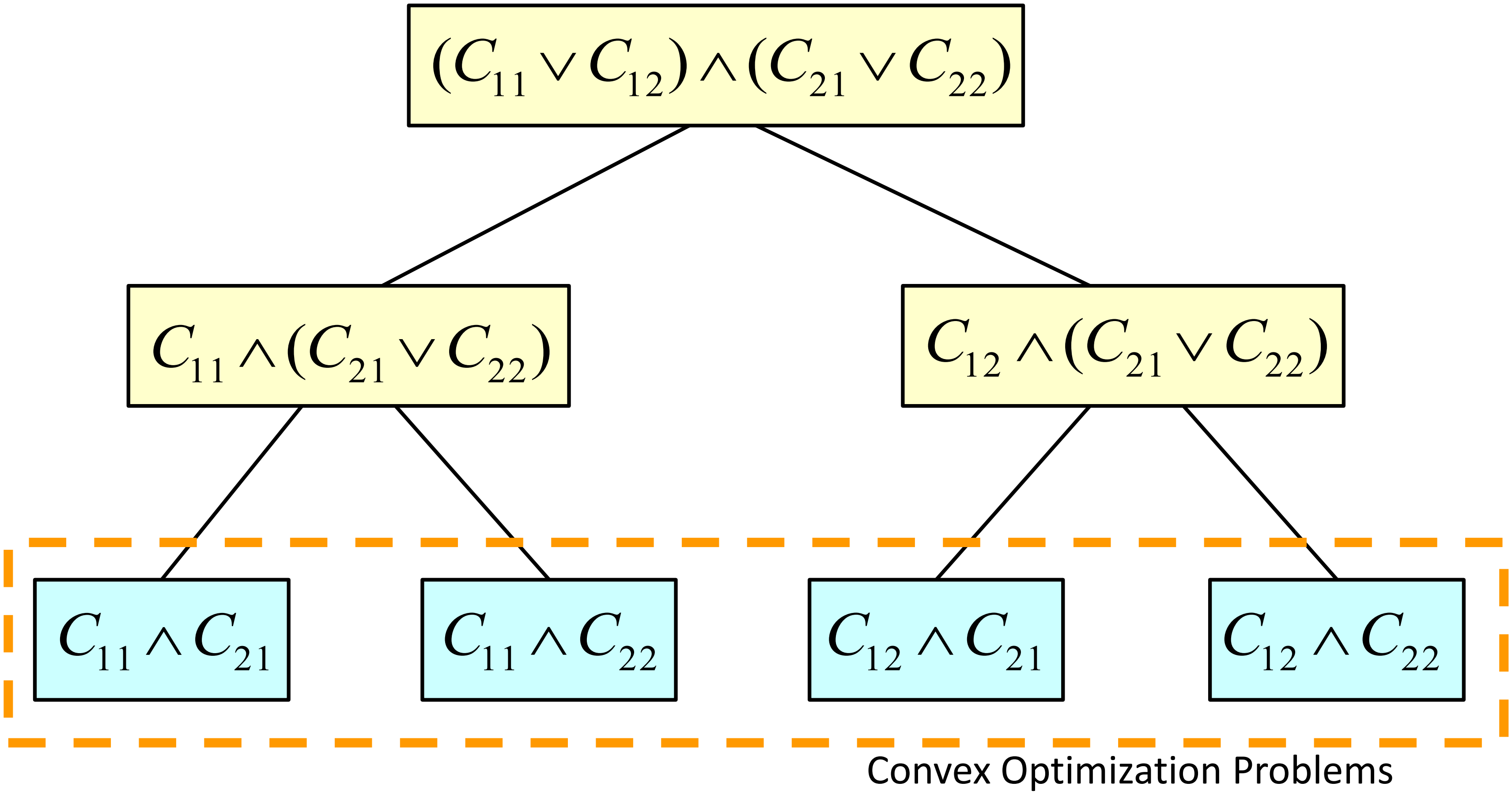
$$s.t. \quad \bigwedge_{t=1}^{T}\bigwedge_{n=1}^{N}\bigvee_{k=1}^{K} h_t^{n,kT}\bar{x}_t \le g_t^{n,k} - m_t^{n,k}\left(\delta_t^n\right)$$

$$\sum_{t=1,n=1}^{T,N}\delta_t^n \le \Delta$$

Stochastic DLP Branch and Bound

Repeat until no clauses left:

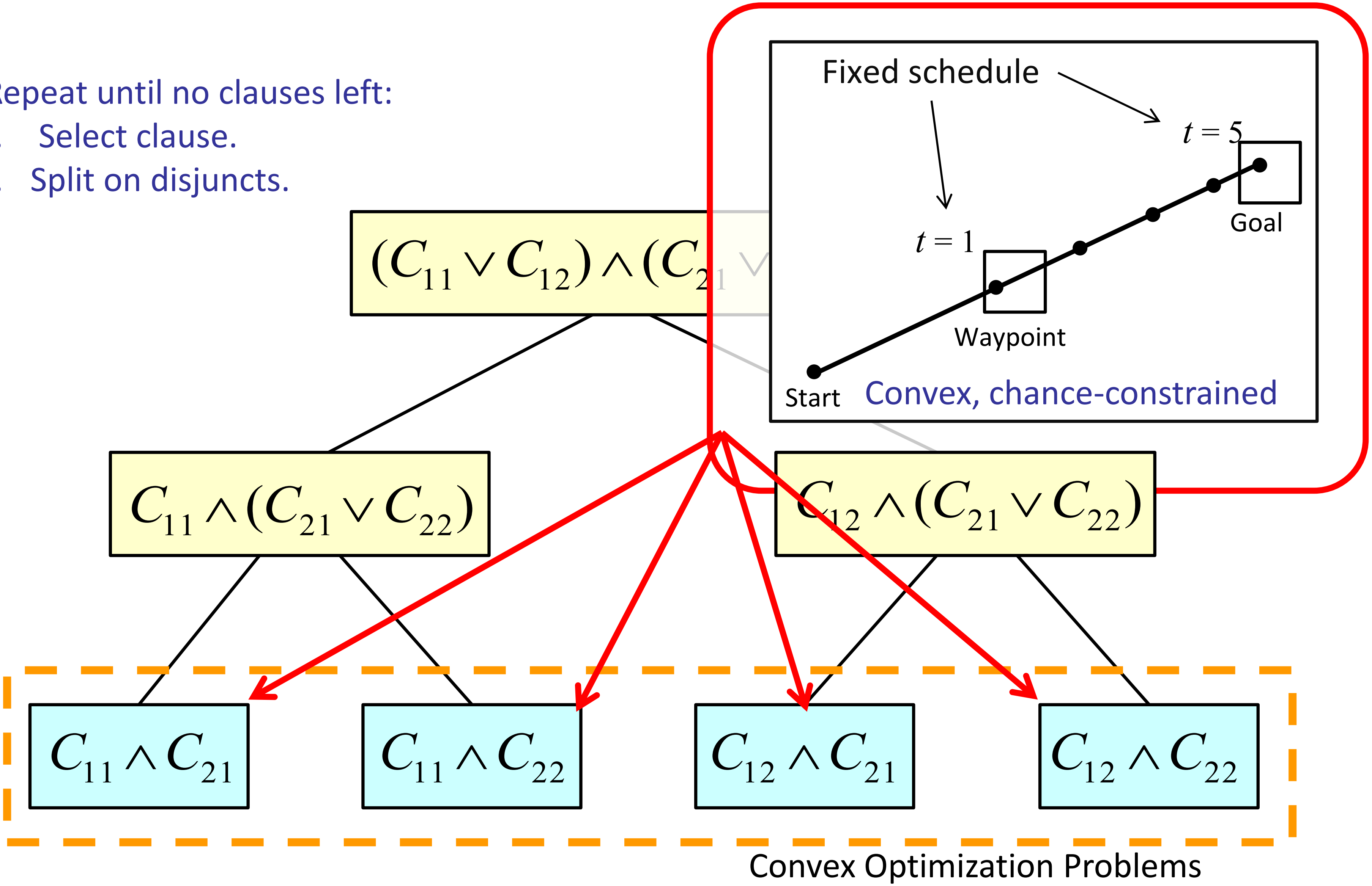
1. Select clause.
2. Split on disjuncts.



Stochastic DLP Branch and Bound

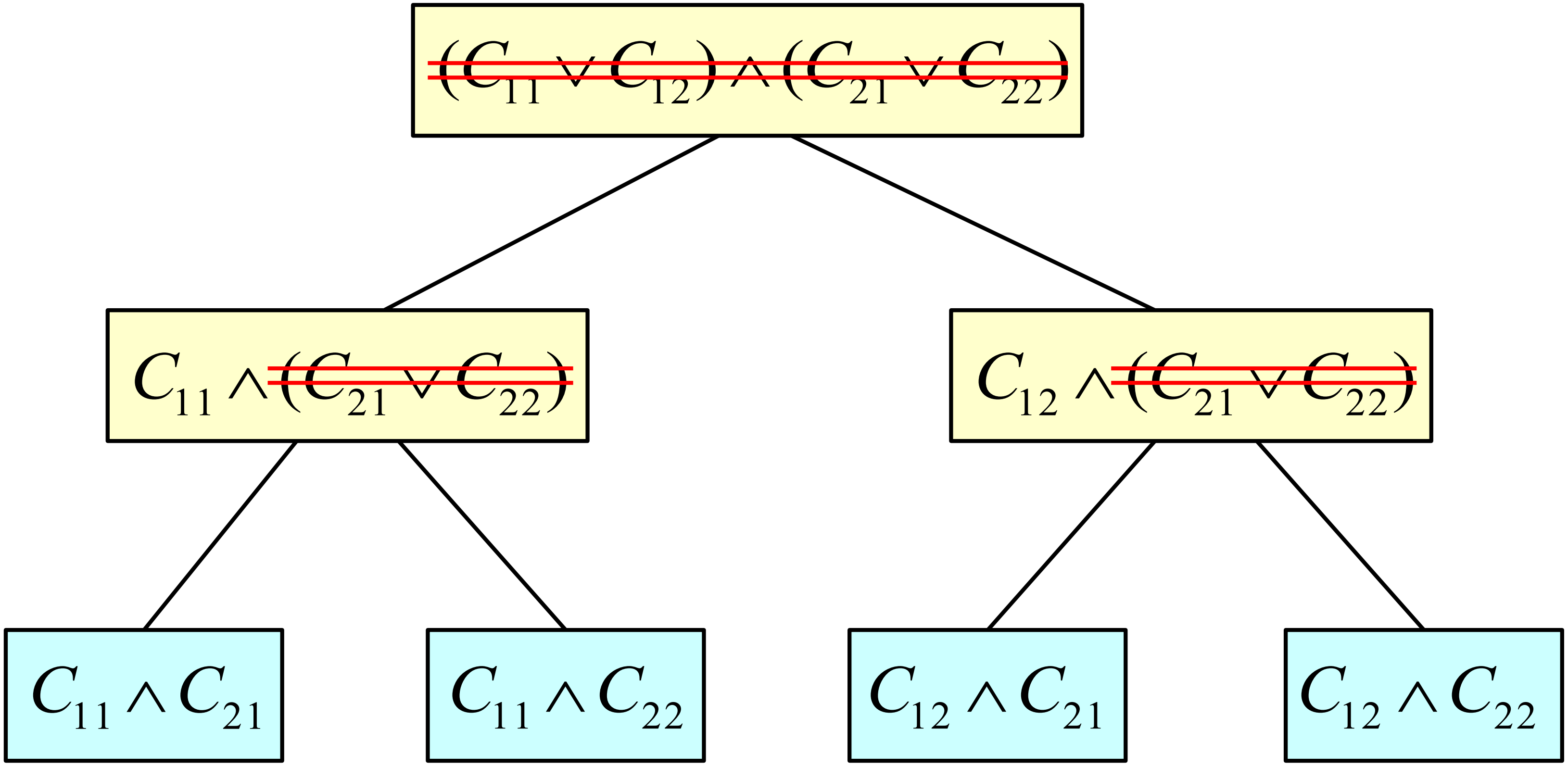
Repeat until no clauses left:

1. Select clause.
2. Split on disjuncts.



Bound Sub-Problems Through Convex Relaxation

- Bound: Remove all disjunctive clauses [Li & Williams 2005].
- Issue: Computing bound is slow!!
- Cause: Sub-problems include non-linear constraints.



B&B Subproblem (non-linear)

$$\min_{u_{1:T}} J(u_{1:T})$$

$$s.t. \quad \forall_{0 \le t \le T-1} \; x_{t+1} = Ax_t + Bu_t + w_t$$

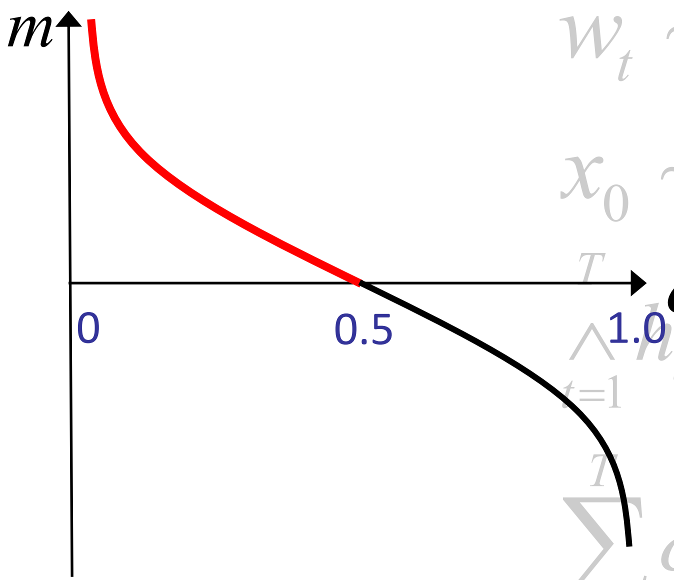
$$w_t \sim N(0, \Sigma_t)$$

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

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

Nonlinear

$$\sum_{t=1}^{T} \delta_t \le \Delta, \; \delta_t \ge 0$$



B&B Subproblem Relaxation (Linear)

$$\min_{u_{1:T}} J(u_{1:T})$$

$$s.t. \quad \forall_{0\le t\le T-1} \; x_{t+1} = Ax_t + Bu_t + w_t$$

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

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

Nonlinear

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

$$\cancel{\sum_{t=1}^{T}\delta_t \le \Delta,}\; \delta_t \ge 0$$

$$\boxed{\text{Fixed Risk Relaxation} \quad \delta_t = \Delta}$$

B&B Subproblem Relaxation (Linear)

$$\min_{u_{1:T}} J(u_{1:T})$$

$$s.t. \quad \forall_{0 \le t \le T-1} \; x_{t+1} = Ax_t + Bu_t + w_t$$

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

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

$$\bigwedge_{t=1}^{T} h_t^{i(t)T} \bar{x}_t \le g_t^{i(t)} - m_t^{i(t)}(\Delta)$$

Constant

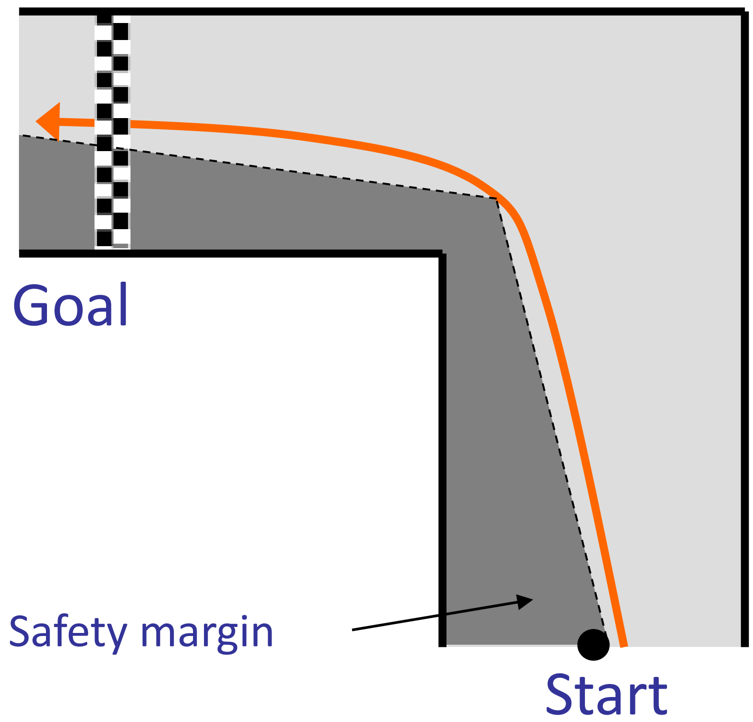
Fixed Risk Relaxation

$$\delta_t = \Delta$$

- All constraints are linear (FRR is typically LP or QP).

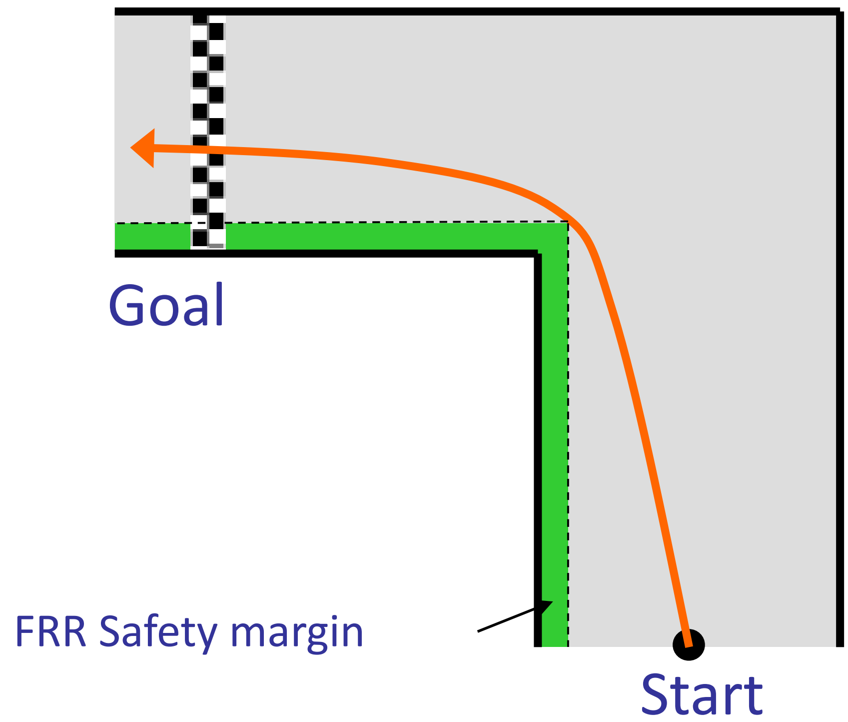
FRR Intuition

Original problem



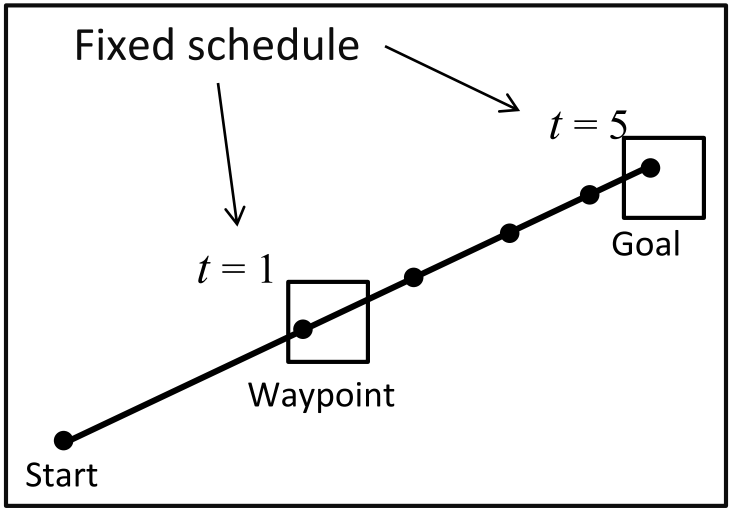
FRR

Sets safety margin for all constraints to max risk Δ.

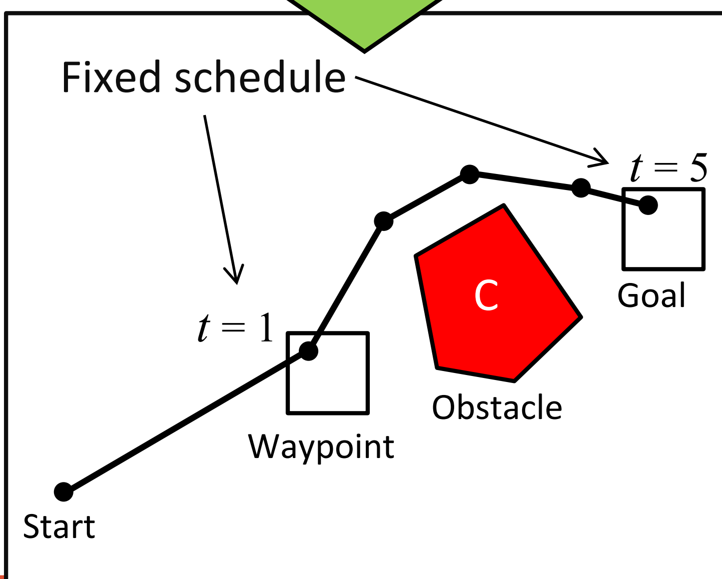


- Results in an **infeasible** solution to the original problem.
- Gives **lower bound** on the cost of the original problem.

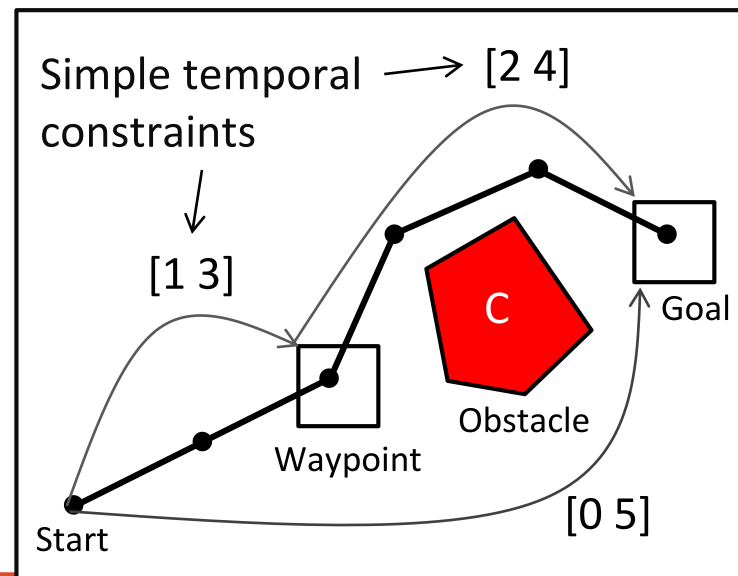
Problems



Convex traj opt



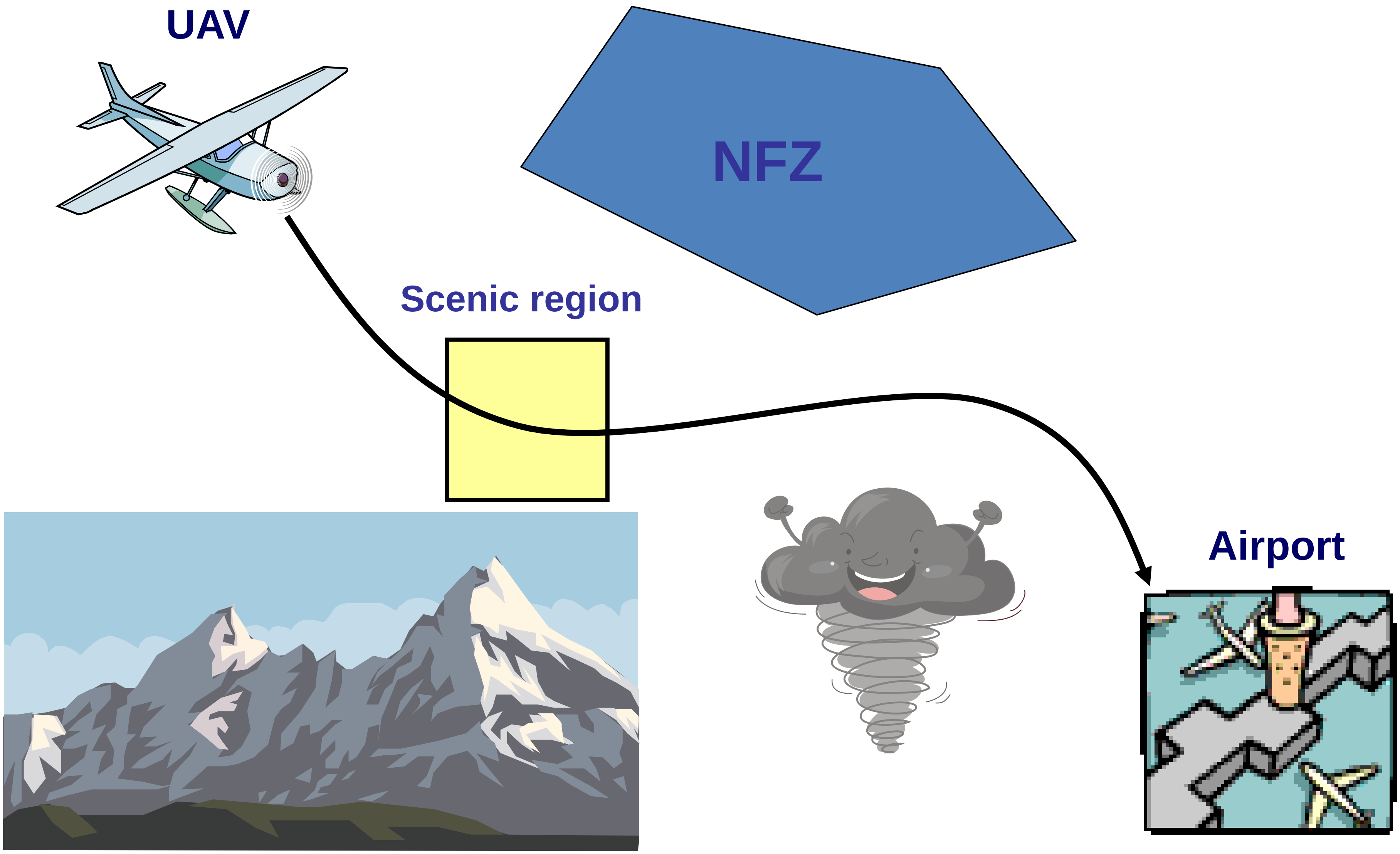
Non-convex, traj opt



Goal-directed qualitative state plan traj opt

APPENDIX: OVERVIEW AND ALTERNATIVE APPROACHES

What are the uncertainties and risks?



Robust Model Predictive Control

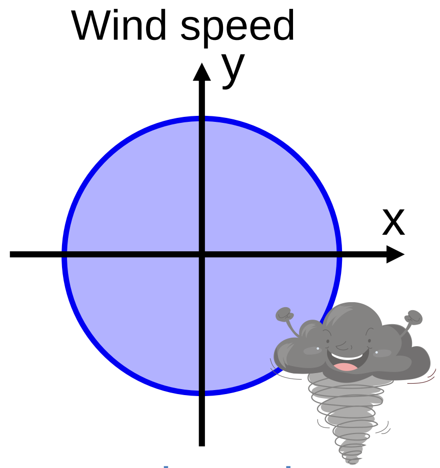
- Receding horizon (MPC) planners react to uncertainty ***after*** something goes wrong.

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

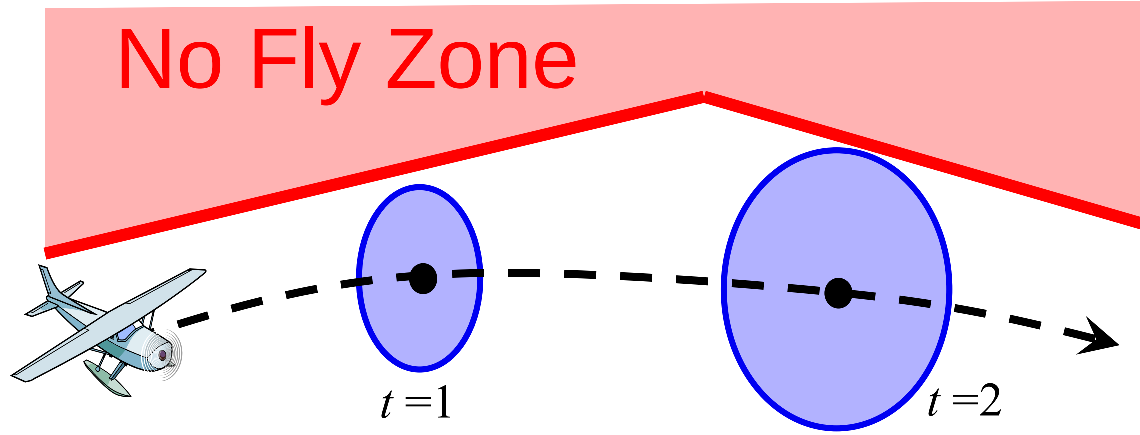
•Ali A. Jalali and Vahid Nadimi, "A Survey on Robust Model Predictive Control from 1999-2006."

Robust versus Chance Constrained

Robust Predictive Control

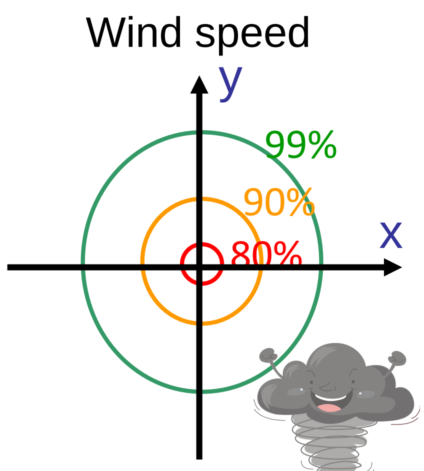


Assume bounds on uncertainty.

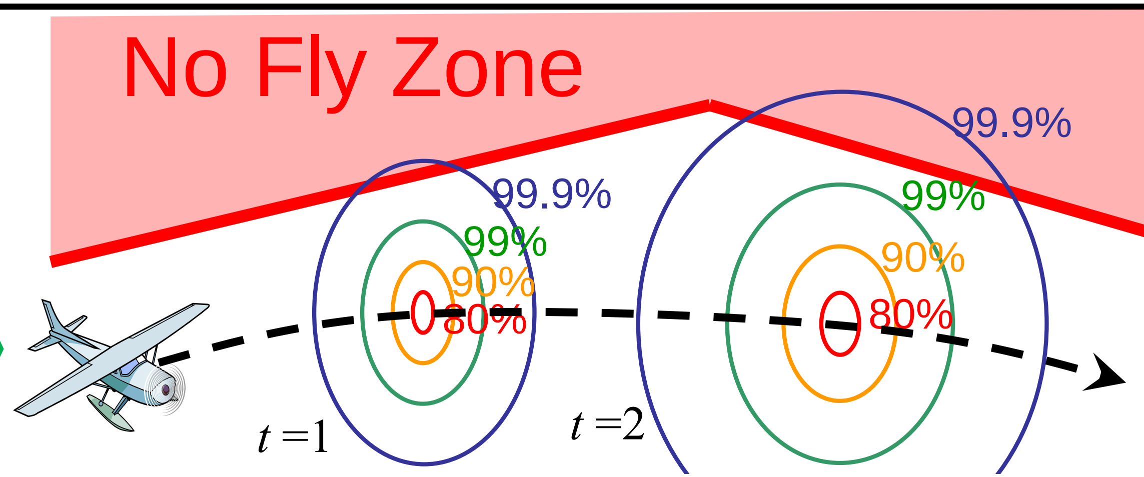


- Predicted position has bounded uncertainty.
- Problem: Find control sequence that satisfies constraints for all realizations of uncertainty.

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*).

Incorporating Uncertainty

- Deterministic discrete-time LTI model.

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

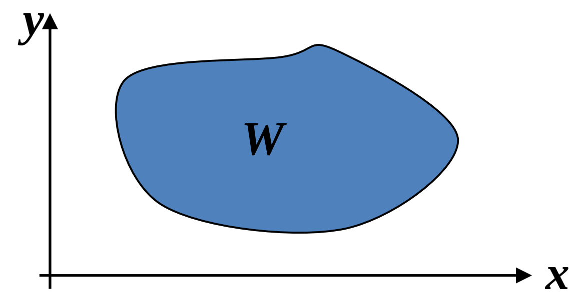
- Additive uncertainty

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

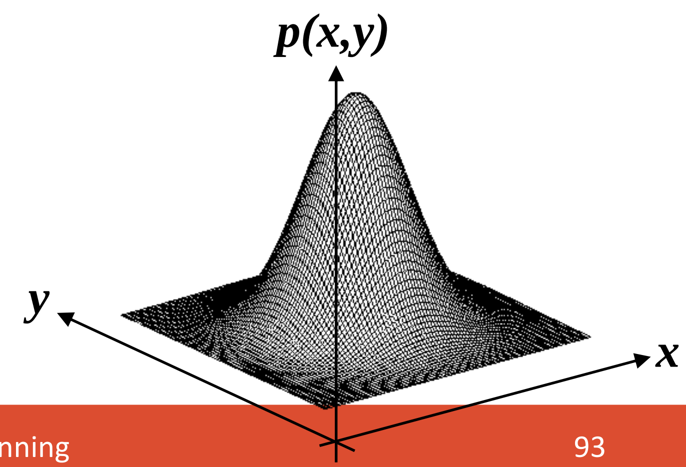
- Multiplicative uncertainty

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

$$w_t \in W$$



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



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

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

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

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

What to Minimize? (*Stochastic* Uncertainty)

- Utilitarian approach

$$\min_{\mathbf{U}} J(\bar{\mathbf{X}}, \mathbf{U}) + p\underline{f(\mathbf{U})}$$

Penalty (constant)

Probability of failure

- **Chance constrained optimization**

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

$$s.t. \quad \underline{f(\mathbf{U})} \leq \underline{\Delta}$$

Probability of failure

Risk bound

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

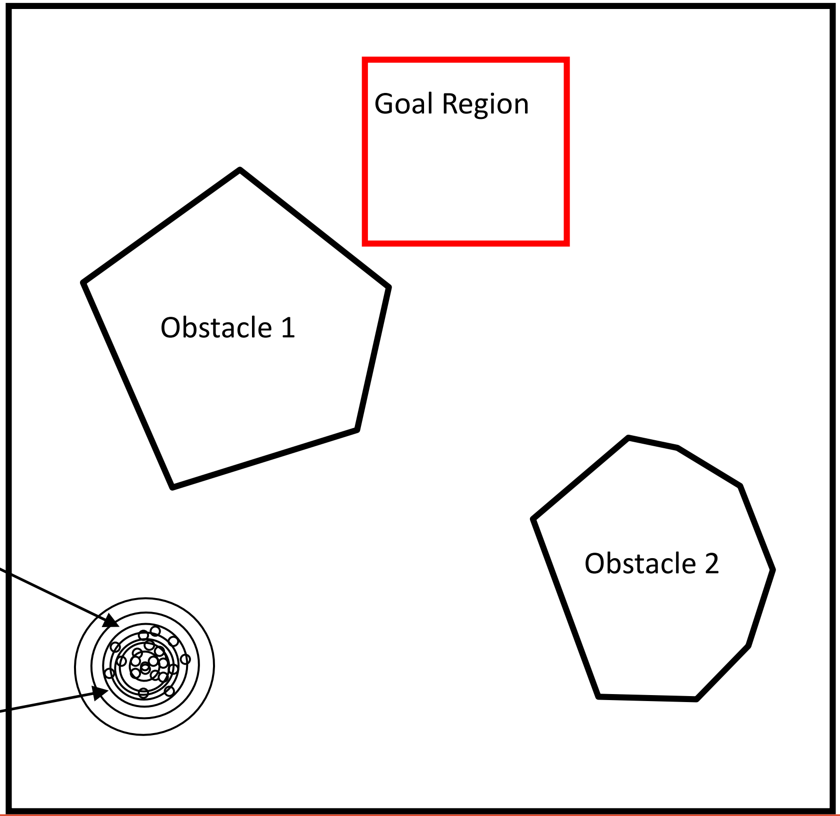
Particle Control

1. Use particles to sample random variables.

$$\mathbf{x}_{c,0}^{(i)} \sim p(\mathbf{x}_{c,0}) \quad v_t^{(i)} \sim p(v_t) \qquad i = 1 \ldots N \quad t = 0 \ldots F$$

Initial state distribution.

Particles approximating initial state distribution.



Particle Control

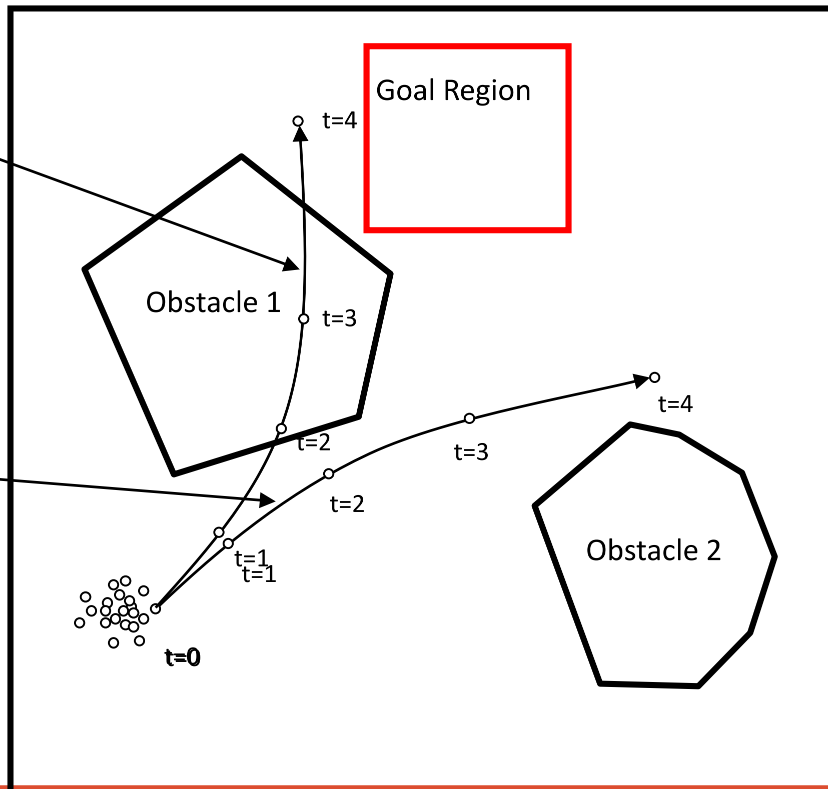
2. Calculate future state trajectory for each particle, **leaving explicit, dependence** on control inputs $\mathbf{u}_{0:\mathrm{T-1}}$.

$$\mathbf{x}_t^{(i)} = f_t(\mathbf{u}_{0:t-1}, \mathbf{x}_{c,0}^{(i)}, \nu_{0:t-1}^{(i)})$$

$$\mathbf{x}_{c,0:T}^{(i)} = \begin{bmatrix} \mathbf{x}_{c,0}^{(i)} \\ \vdots \\ \mathbf{x}_{c,F}^{(i)} \end{bmatrix}$$

Particle 1 for $\mathbf{u} = \mathbf{u}_{\mathrm{B}}$

Particle 1 for $\mathbf{u} = \mathbf{u}_{\mathrm{A}}$

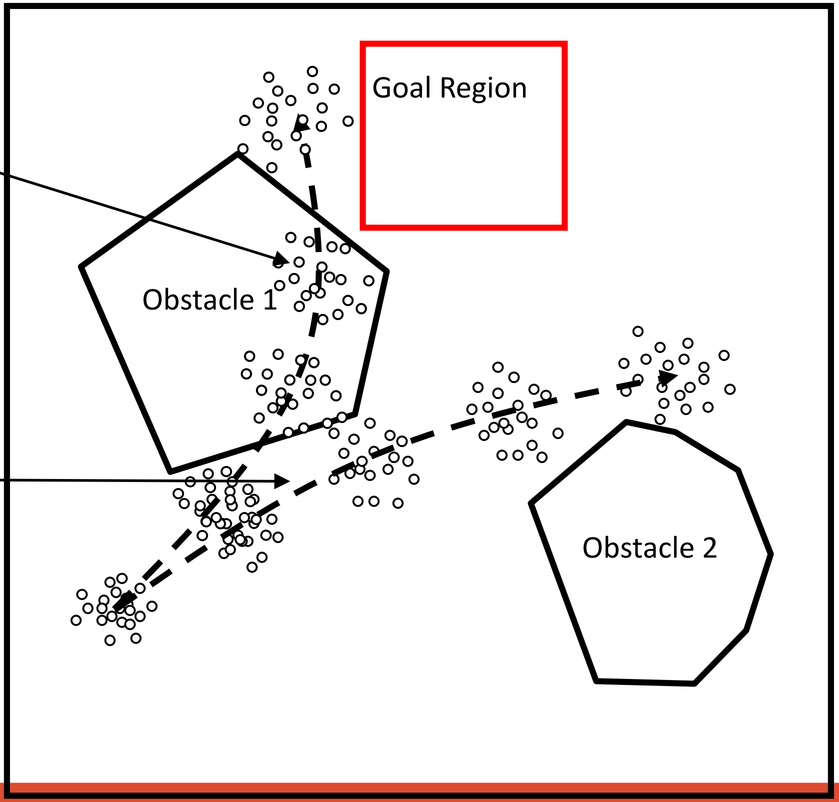


Particle Control

2. Calculate future state trajectory for each particle, **leaving explicit, dependence** on control inputs $\mathbf{u}_{0:T\text{-}1}$.

Particles $1 \ldots N$
for $\mathbf{u} = \mathbf{u}_B$

Particles $1 \ldots N$
for $\mathbf{u} = \mathbf{u}_A$



Particle Control

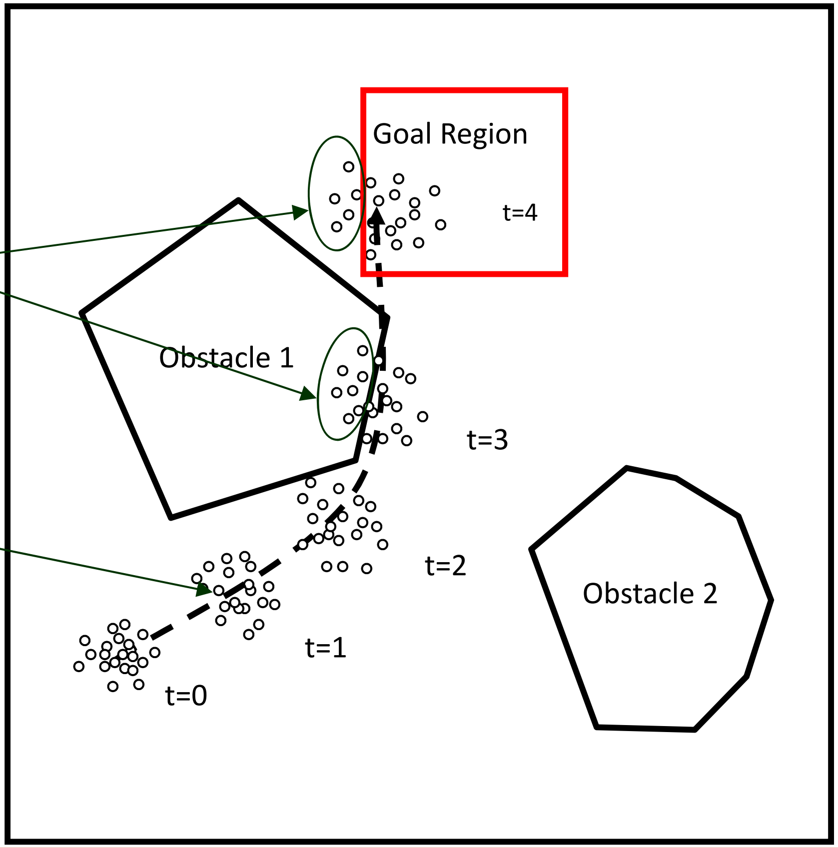
3. Express **chance-constraints of** optimization problem **approximately** in terms of particles.

LB4

Probability of failure approximated by the **fraction** of failing particles.

.

Sample mean approximates **state mean.**



True expectation approximated by sample mean of cost function:

$$E\left[h(\mathbf{u}_{0:F-1}, \mathbf{x}_{1:F})\right]$$

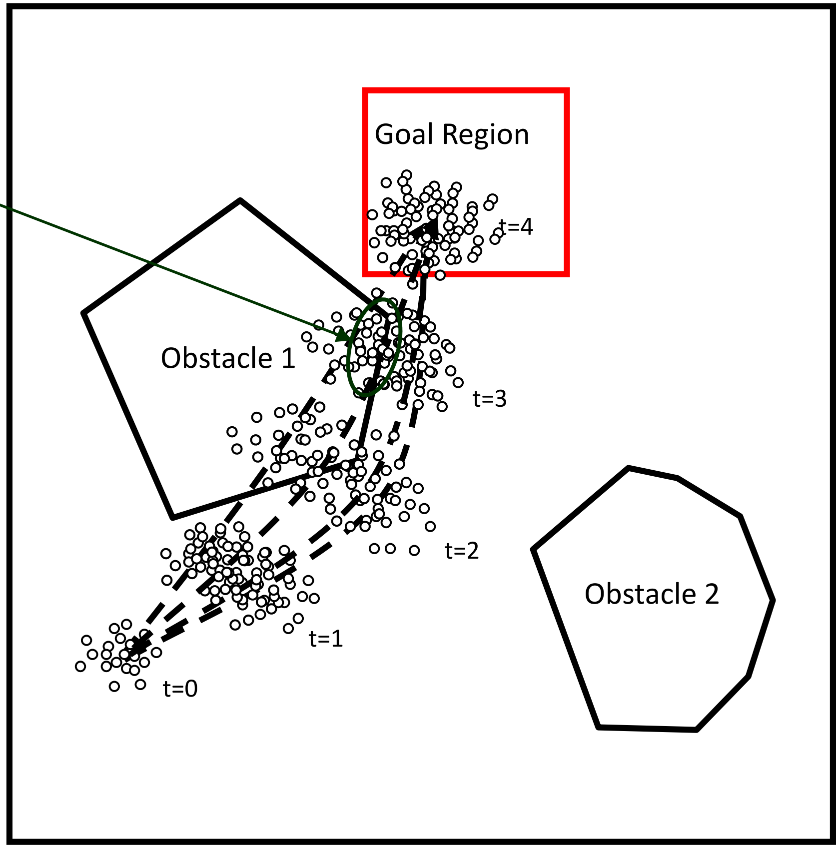
$$\approx \frac{1}{N}\sum_{i=1}^{N} h(\mathbf{u}_{0:F-1}, \mathbf{x}_{1:F}^{(i)})$$

Particle Control

4. Solve approximate **deterministic** optimization problem for $\mathbf{u}_{0:F-1}$.

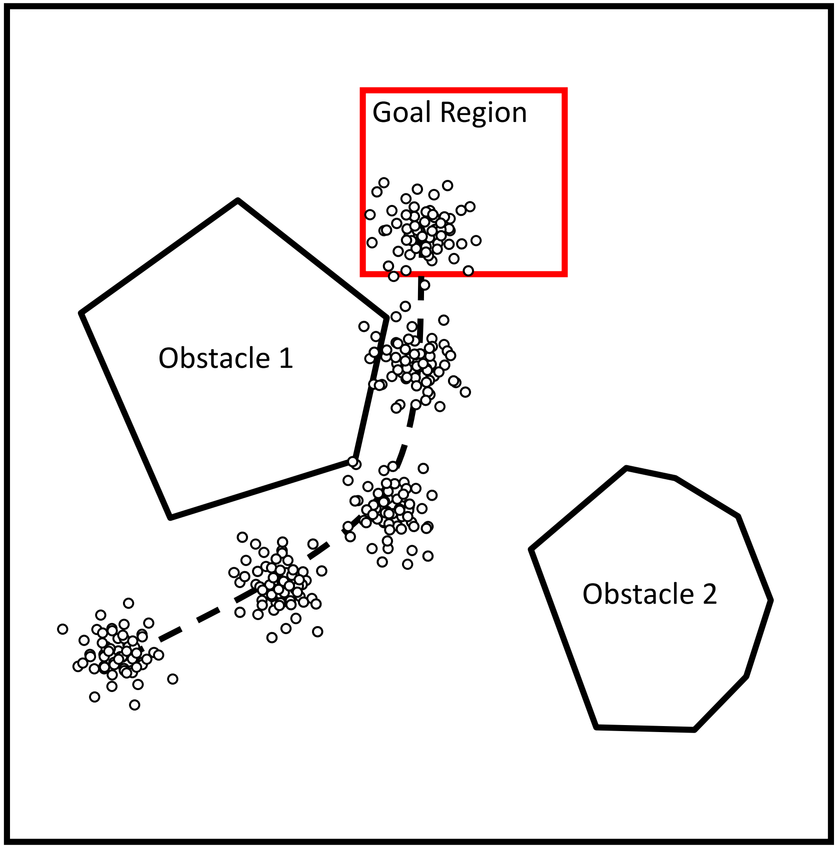
$\delta = 0.1$

10% of particles fail in optimal solution.



Convergence

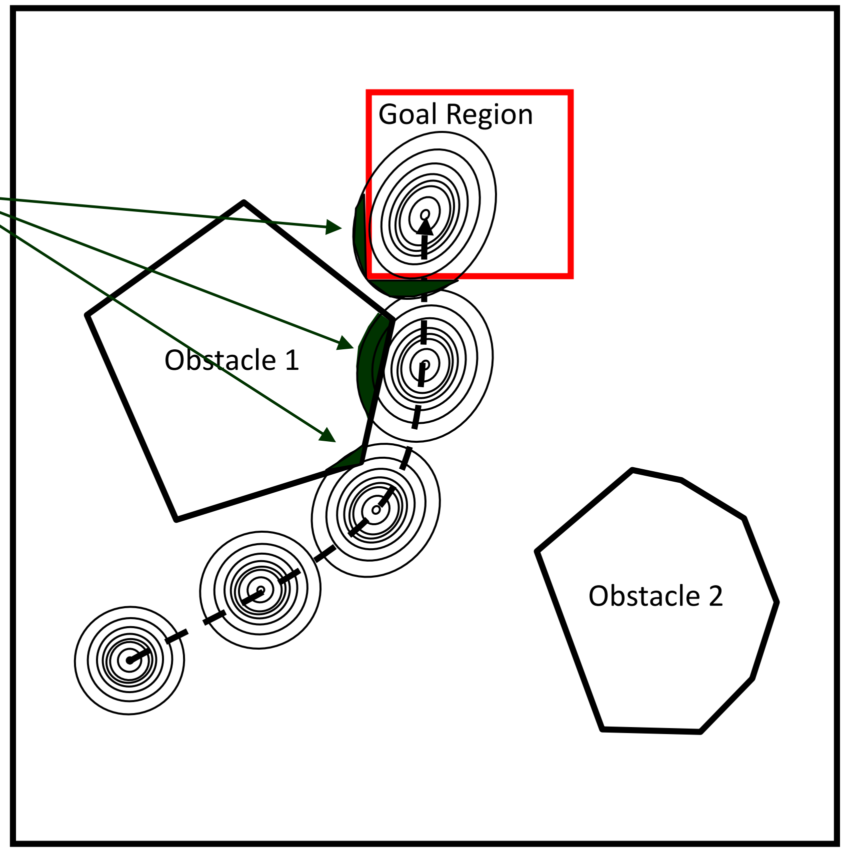
– As N→∞, approximation becomes exact.



Convergence

– As N→∞, approximation becomes exact.

10% probability of failure.



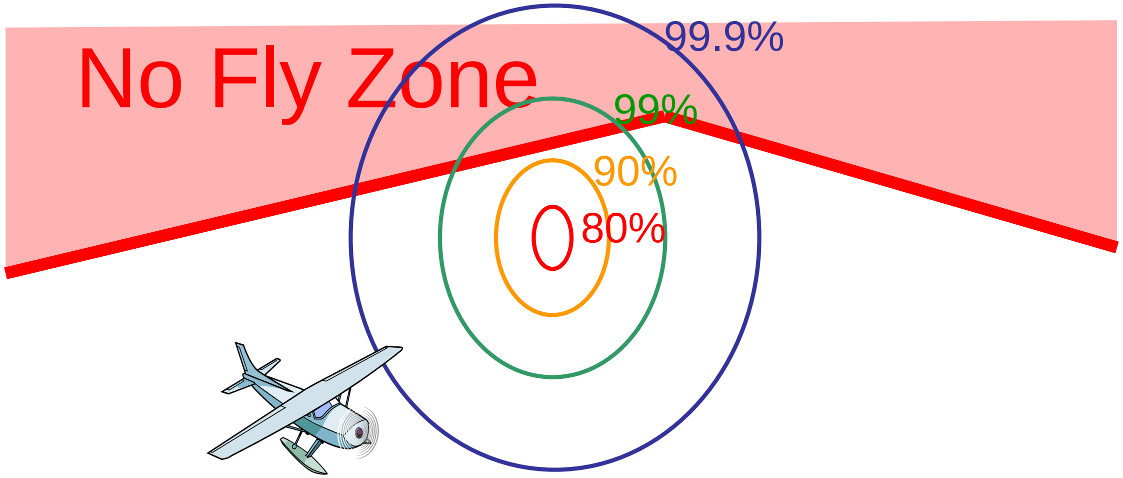
Solution Methods for Chance-Constrained Problems

- 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

Elliptic Approximation

Chance constraint:

Risk < 1%



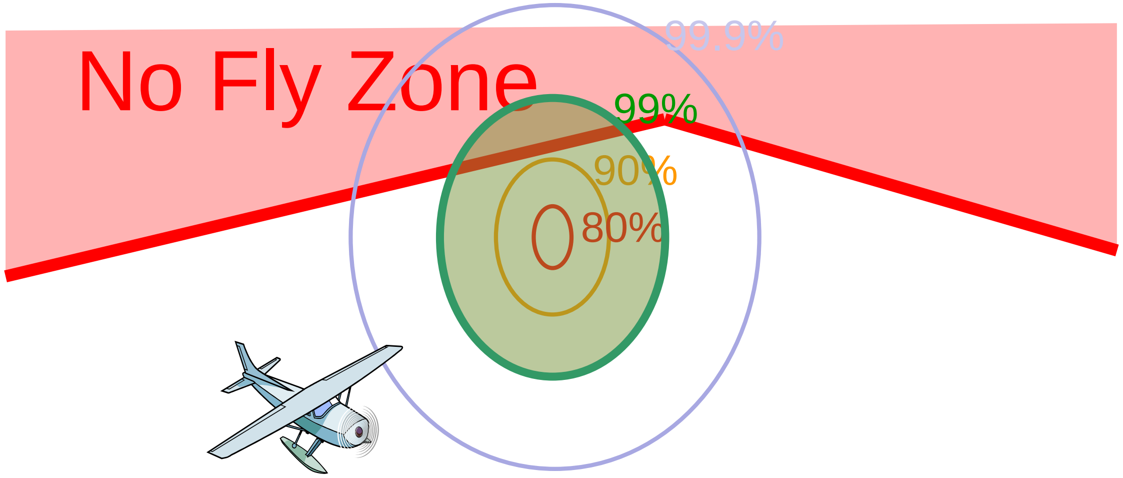
1. Derive probability distribution over future states as a function of control inputs.

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%

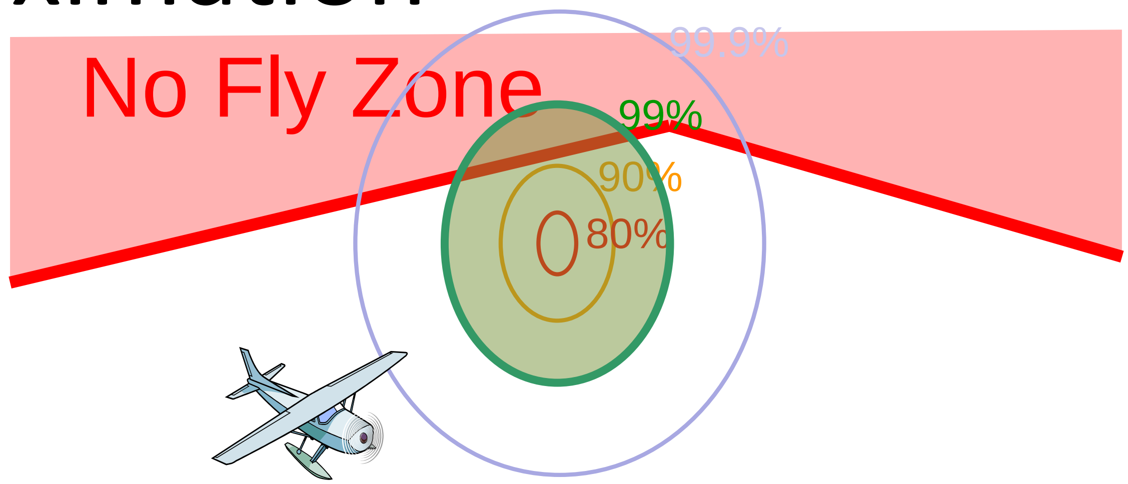


1. Derive probability distribution over future states as a function of control inputs.
2. Find a 99% probability ellipse.

Elliptic Approximation

Chance constraint:

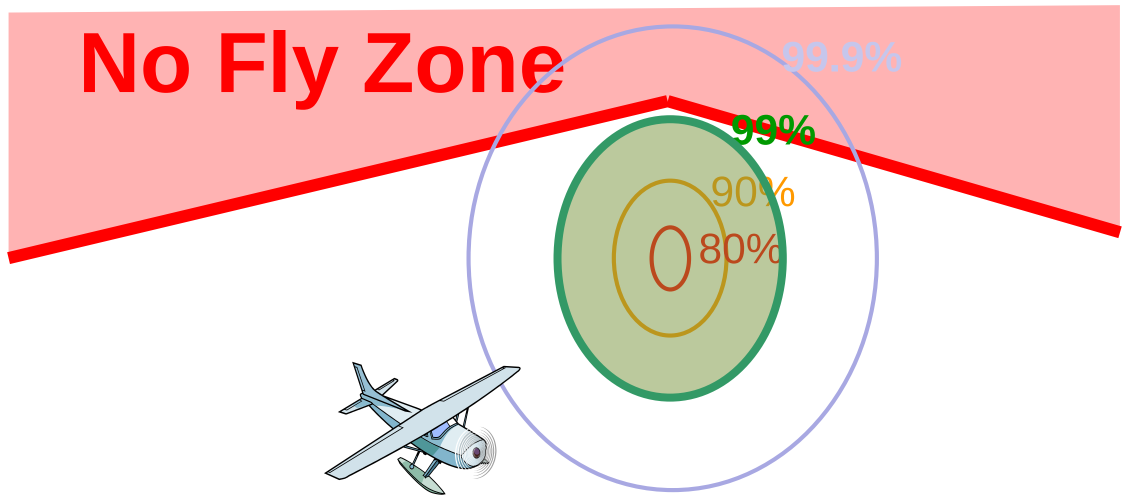
Risk < 1%



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

Conservatism of Elliptic Approximation

Issue: often *very* conservative



Real probability of failure

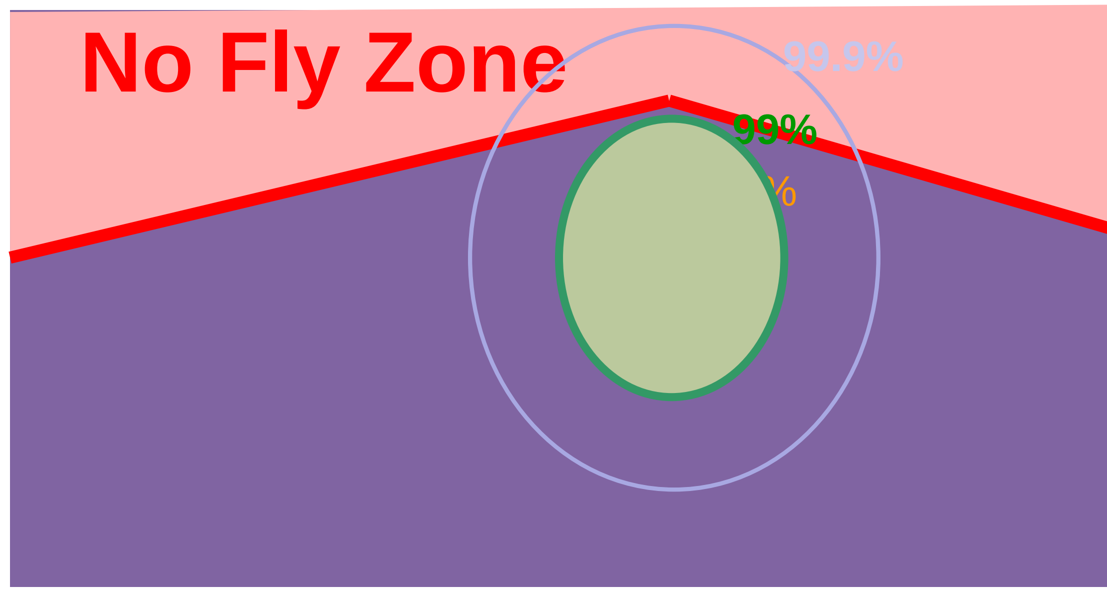
$$= \int p(x)dx$$

Probability density function

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

Conservatism of Elliptic Approximation

Issue: often *very* conservative.



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