Massachusetts Institute of Technology

Stochastic Optimization and Risk Allocation

CPAIOR Master's Tutorial

May 18th, 2013

Brian C. Williams (MIT) and
Masahiro Ono (Caltech JPL)

Joint work with
Lars Blackmore (Space X)

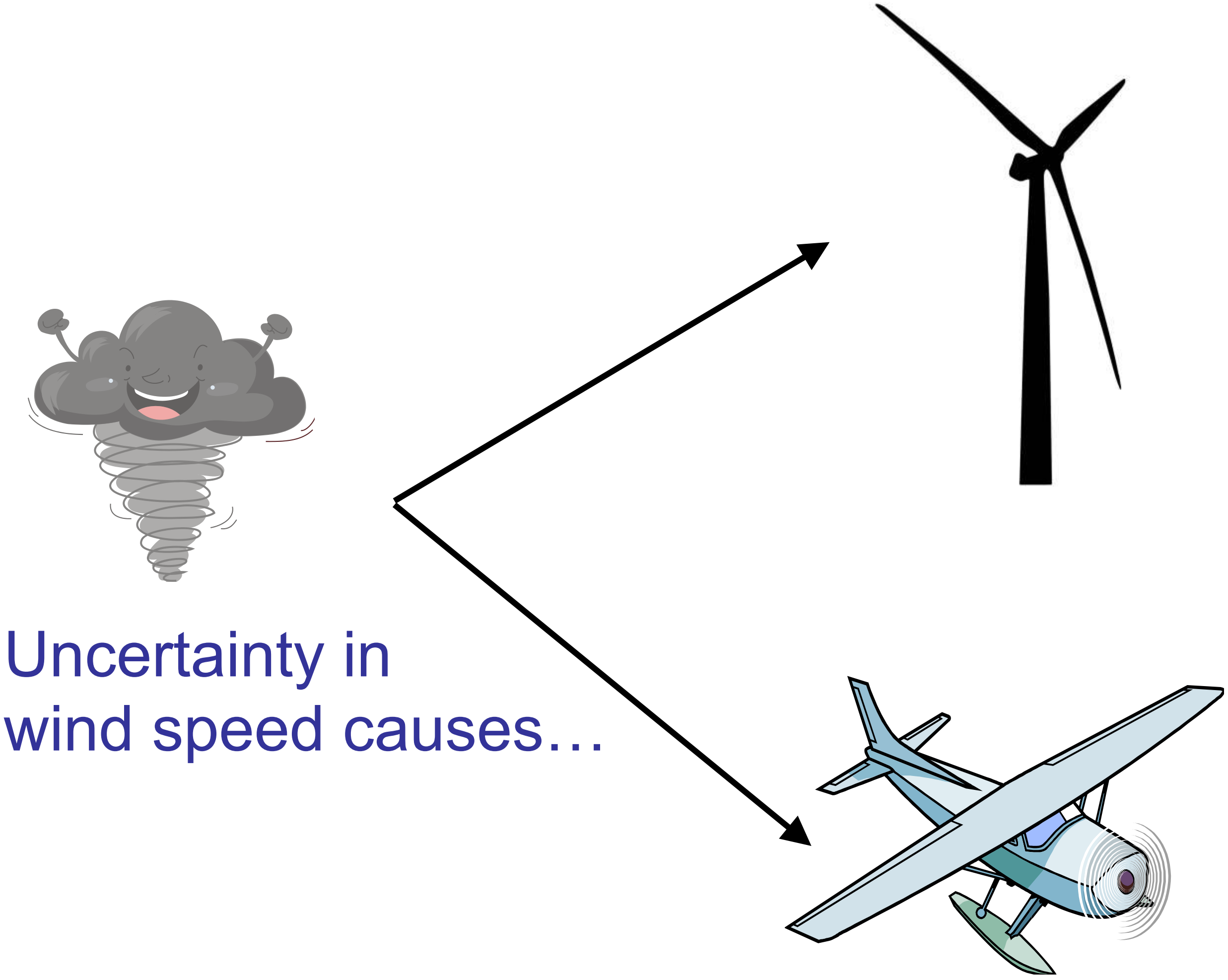
Model-based Embedded & Robotic Systems

CSAIL

The world is uncertain.

Some levels of risk are unacceptable.

Impact of Uncertainty on Dynamic Systems



Uncertainty in wind speed causes…

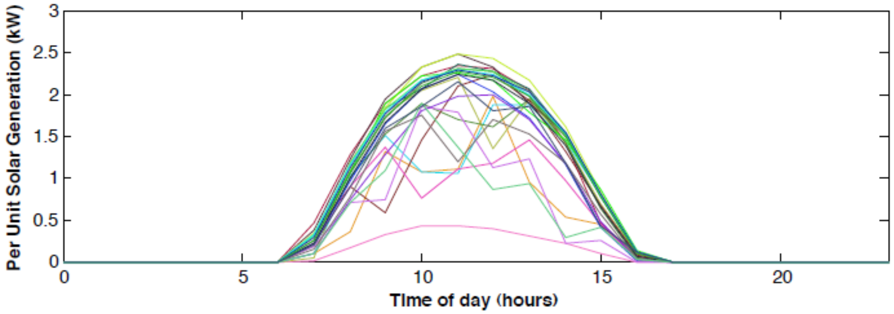
Uncertainty in wind power

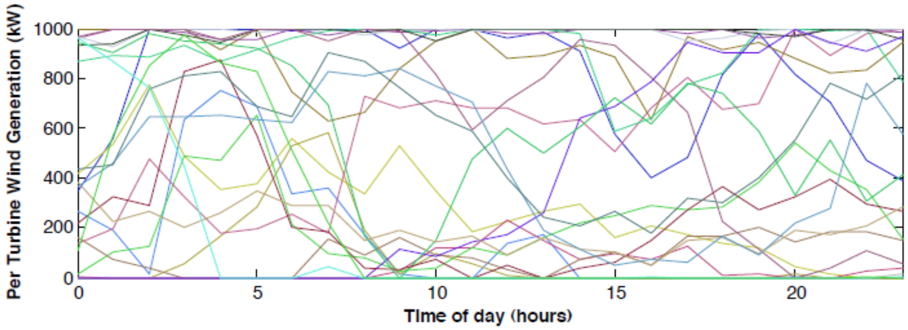
Uncertainty in vehicle position

Motivation: High Penetration of Renewables

Electrical grid must prepare for high penetration of renewables.

Challenge: Wind and solar are undispatchable, intermittent, and unpredictable.



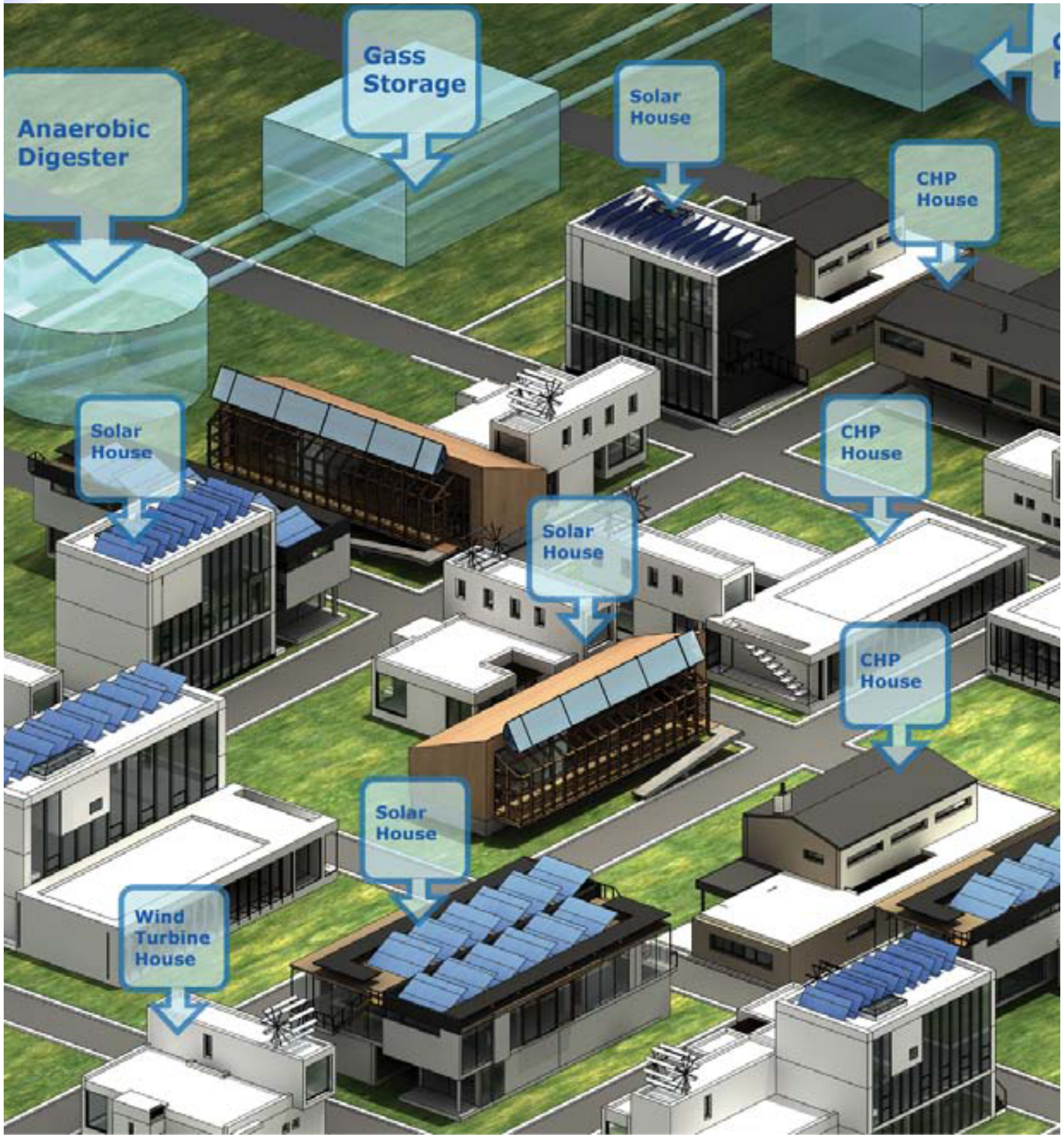


Sustainable Homes

- Goal: Optimally control HVAC, window opacity, washer/dryer, e-car.
- Objective: Minimize energy cost.
- Uncertainty: Solar input, outside temp, energy supply, occupancy.
- Risk: Resident goals not satisfied; occupant uncomfortable.

Connected Sustainable Homes Testbed
Federico Casalegno (PI), MIT Mobile Experience Lab

(Sub)Urban Scale Sustainability



- Heterogeneous connected homes with different energy sources.
- Symmetric energy exchange between houses.
- Challenge:
  - How to distribute energy optimally,
  - while limiting the risk of an energy shortage,
  - without centralized control.

Bottom-Up Grid Project, MIT-EI-Tata.

Vehicle Electrification and Autonomy

- Barrier to adoption due to range anxiety.

Vehicle Electrification and Autonomy

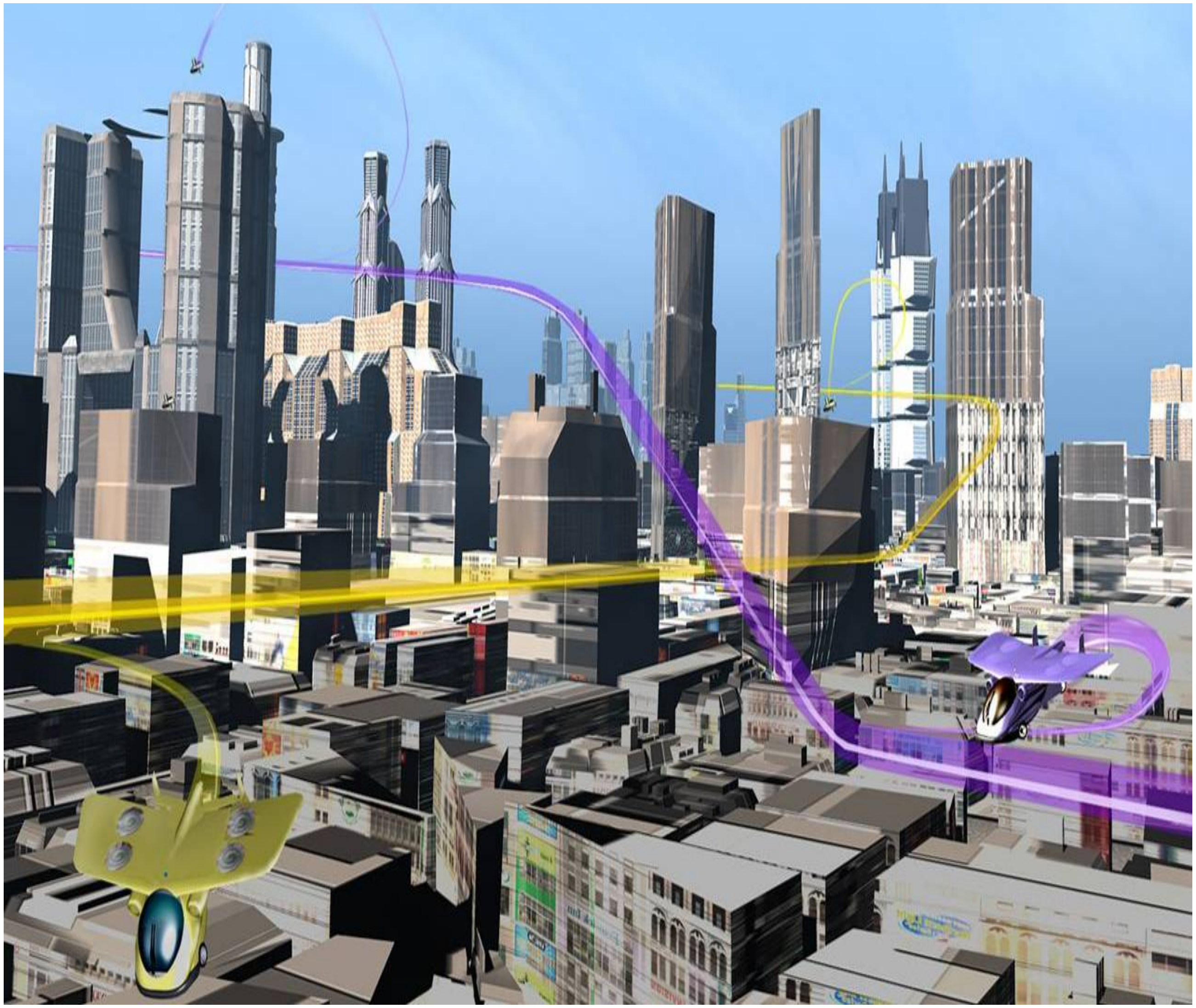
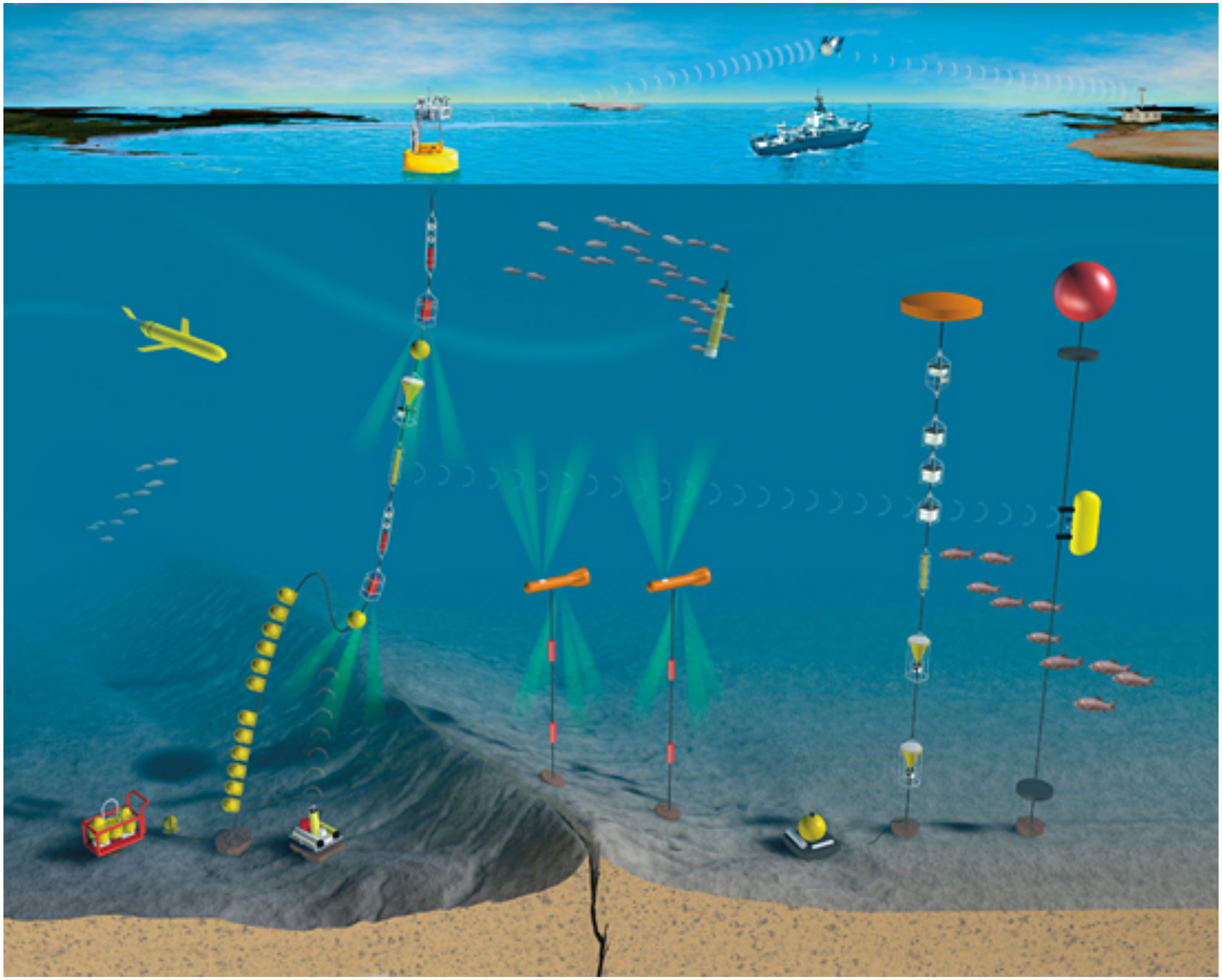


Image courtesy of Boeing Research & Technology

Environmental Observing Systems

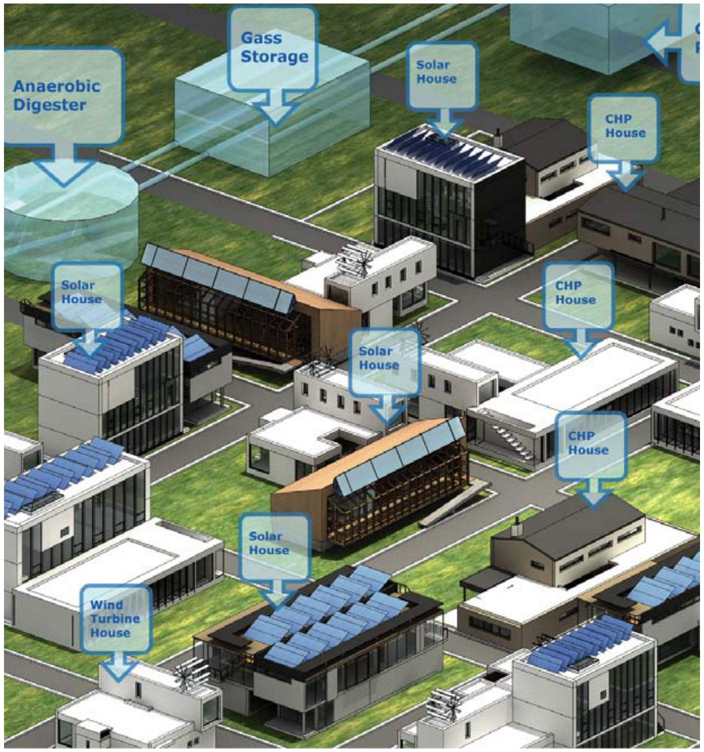


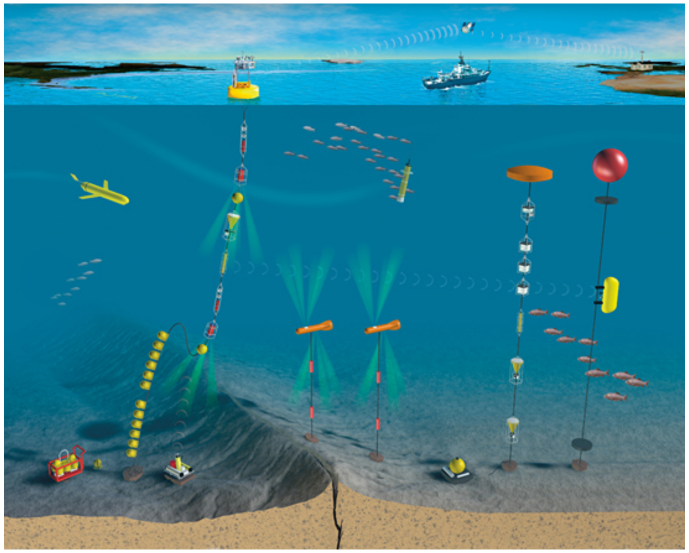
Environmental Observing Systems

- Barriers to high science return include operational cost and mission risk.

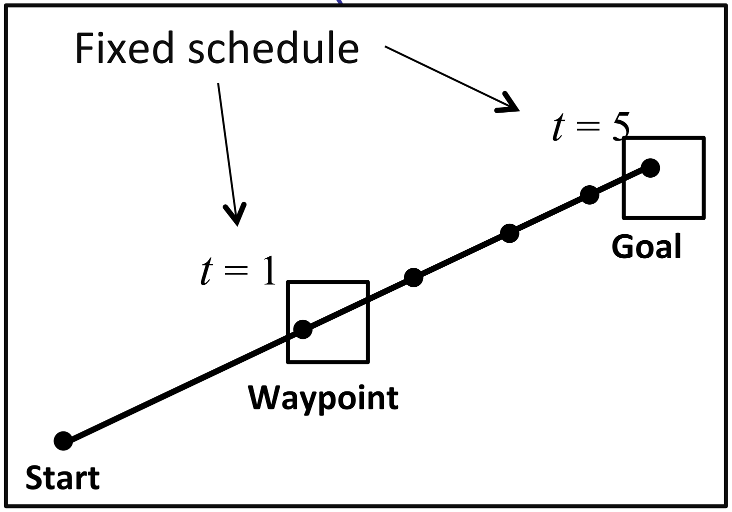
Joint collaboration with
Woodshole Deep Submergence Lab and
The Monterrey Bay Aquarium Research Institute

Facilitating Sustainability Requires Managing Risk

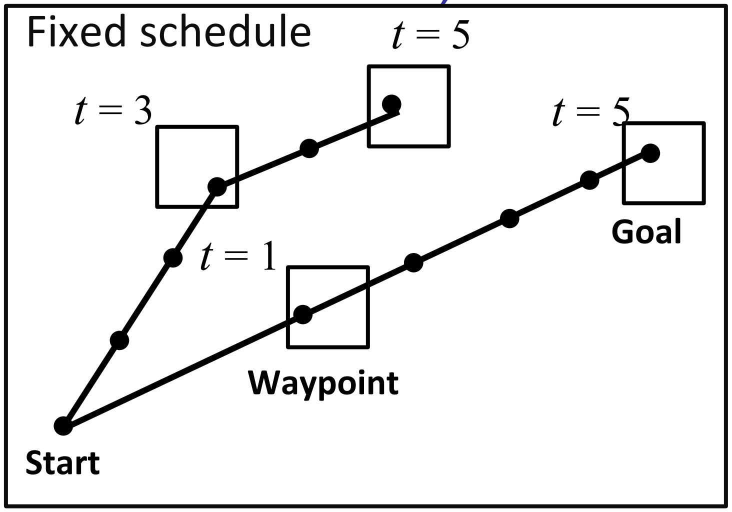




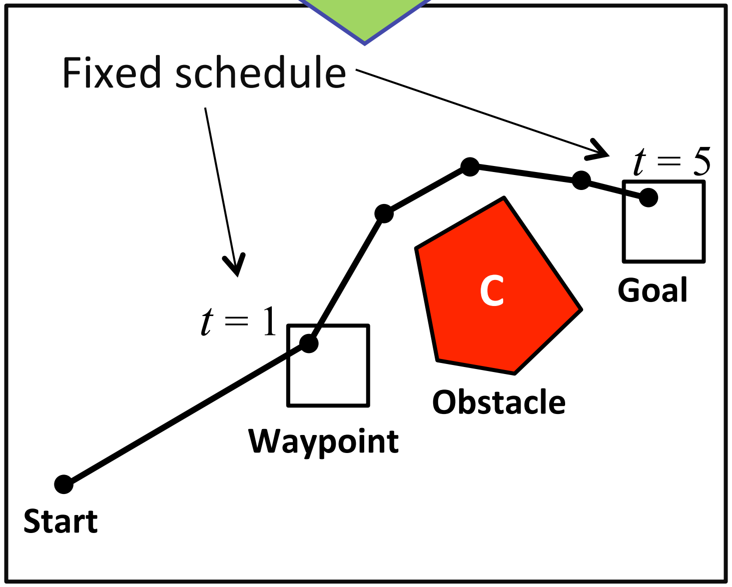
Risk-bounded Planning
(Goal-directed Model-Predictive Control)



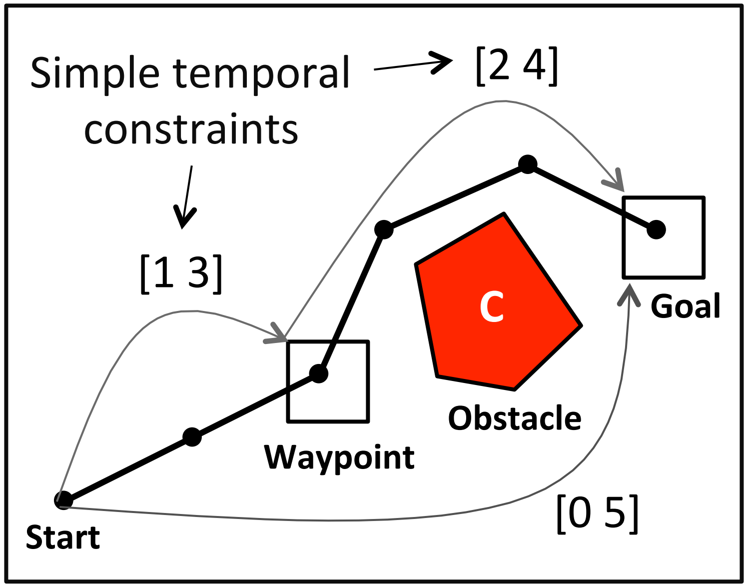
Convex, single agent



Convex, multi-agent



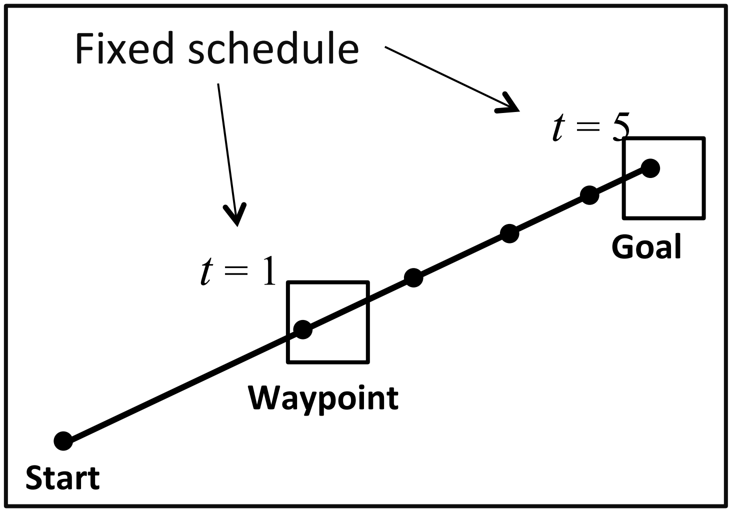
Non-convex, single agent



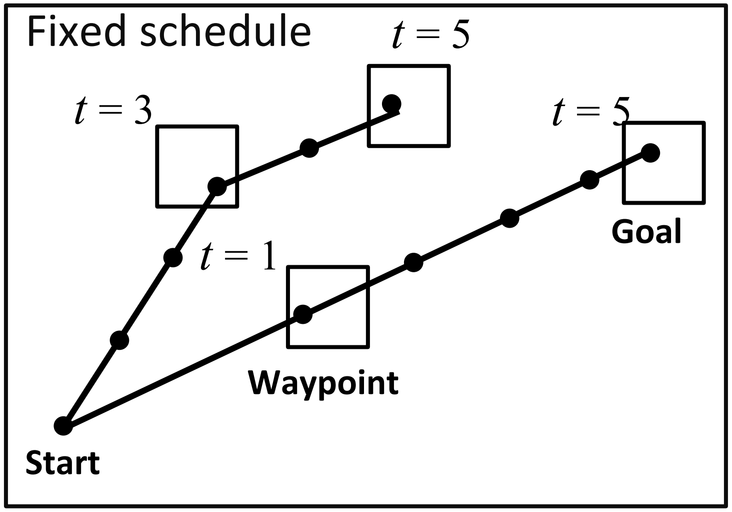
Non-convex, flexible schedule, single agent

Model-based Embedded & Robotic Systems

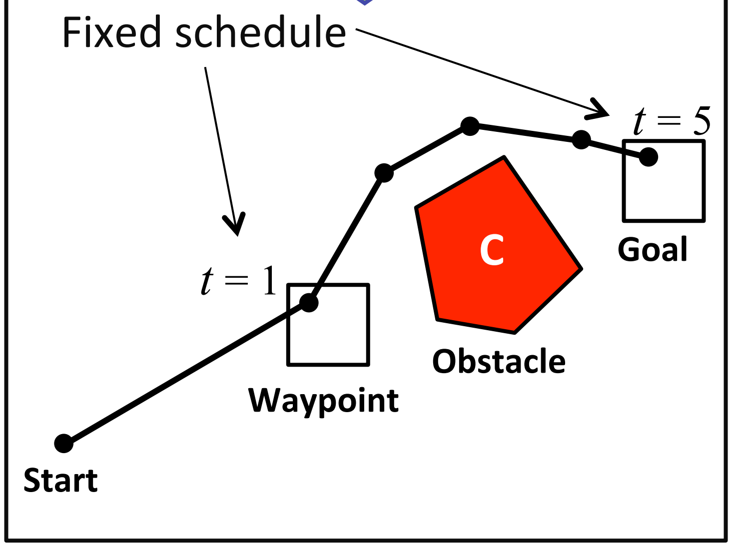
Stochastic Optimization Problems



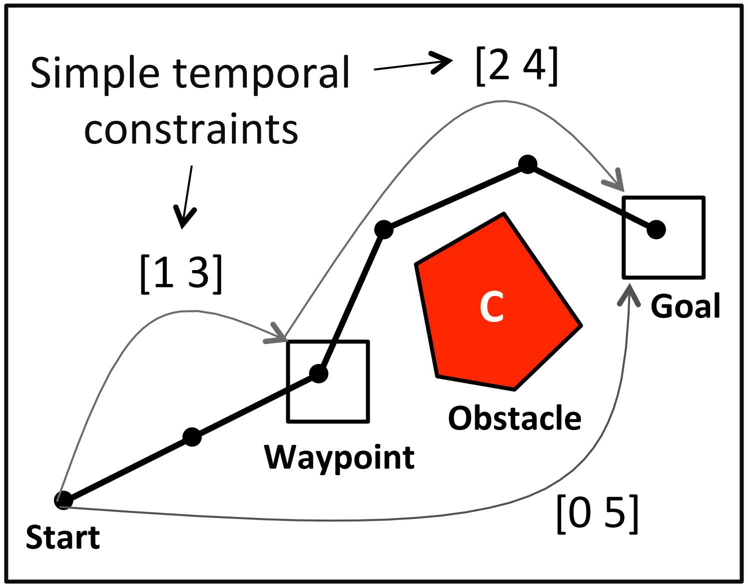
Convex chance-constrained opt.



Decentralized chance-constrained opt

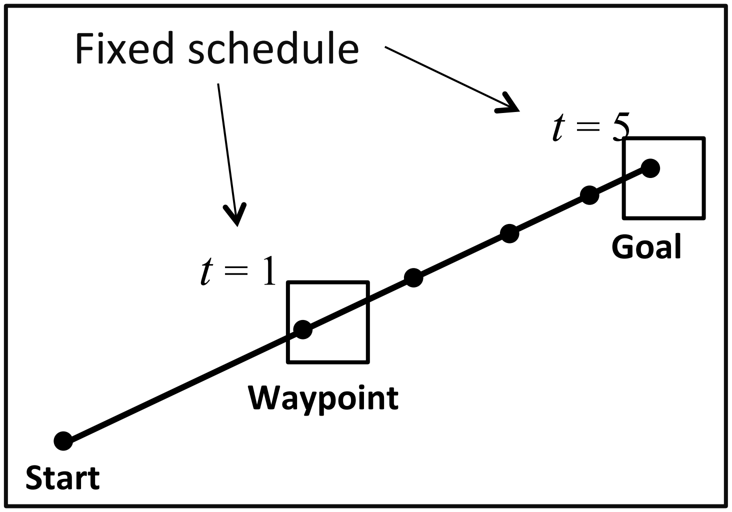


Non-convex, chance-constrained opt

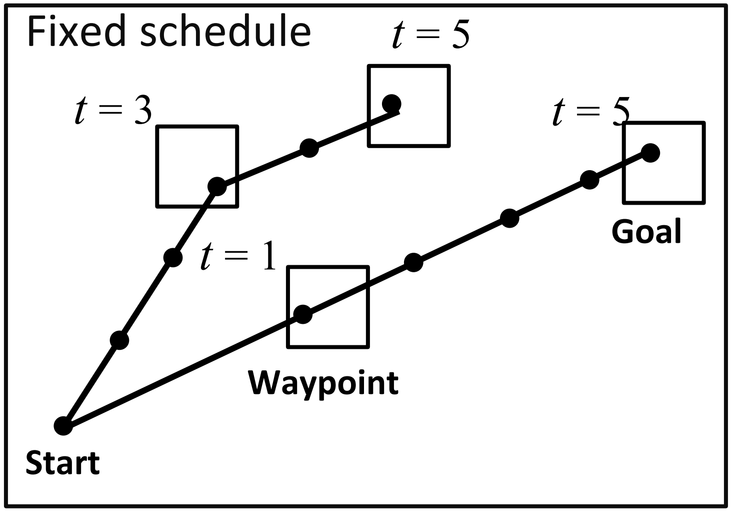


Chance-constrained planning & scheduling

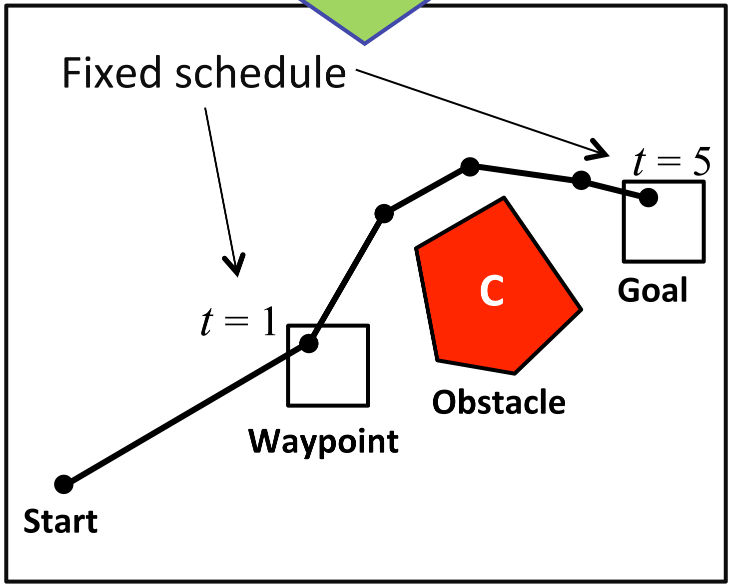
Risk Allocation Algorithms



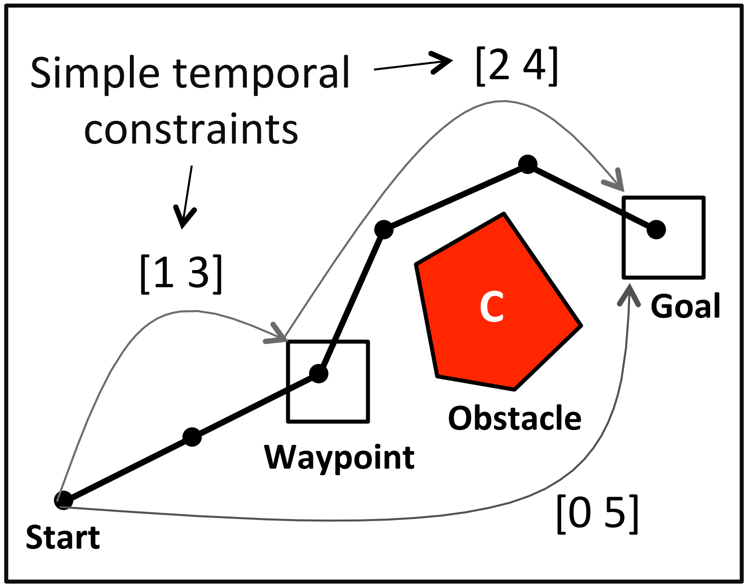
IRA (Iterative Risk Allocation)



MIRA (Market-based IRA)



IRA (Non-convex Risk Allocation)

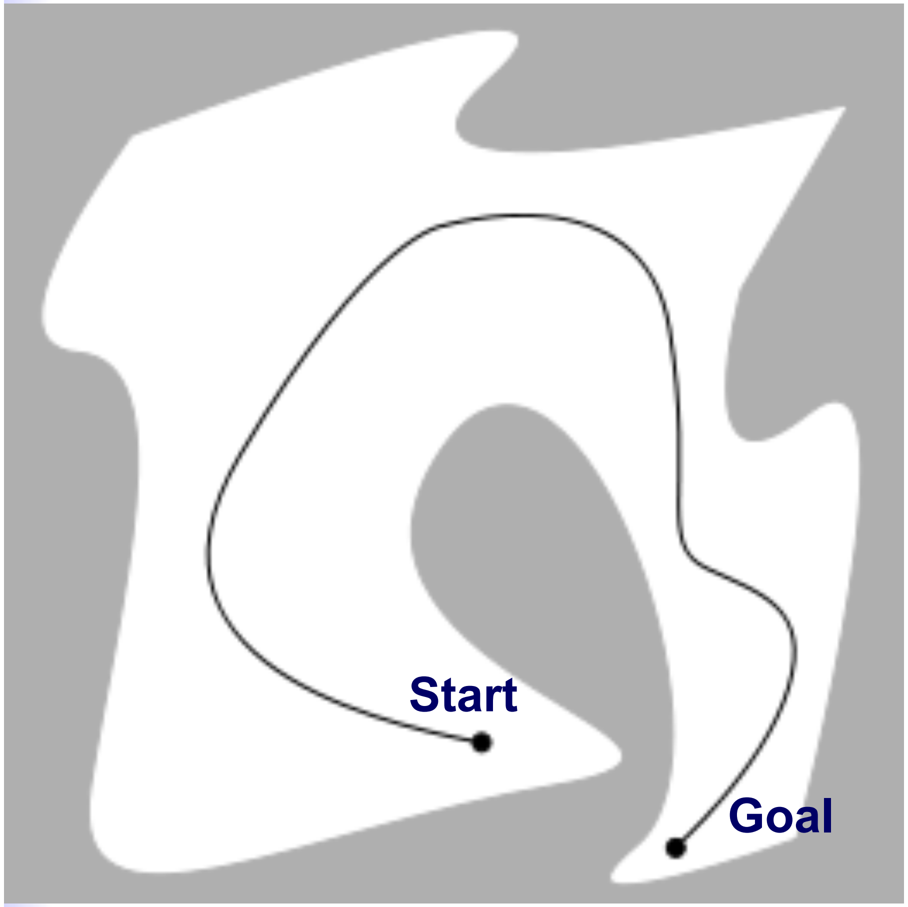


p-Sulu (probabilistic Sulu)

Outline

- Goal-directed, Model-Predictive Control
- Stochastic Optimization
- Iterative Risk Allocation
- Optimal Risk Allocation
- Appendix: Multi-agent Risk Allocation

Model-Predictive Control



- Plan control trajectory = constraint optimization

$$\min_{p} J(p)$$

$$s.t.$$

$$p \in P$$

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

Finite Horizon Model-Predictive Control

- Formulate as Linear (LP), Mixed Integer (MILP) or Mixed-Logic (MLLP) Program.

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

Cost function

$$s.t.$$

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

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

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

Goal position and velocity

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

Actuation limits

$$\mathbf{x}_k \equiv (x_k \quad y_k \quad \dot{x}_k \quad \dot{y}_k)^T, \quad \mathbf{u}_k \equiv (F_{x,k} \quad F_{y,k})^T$$

Example Constraints

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

Dynamics

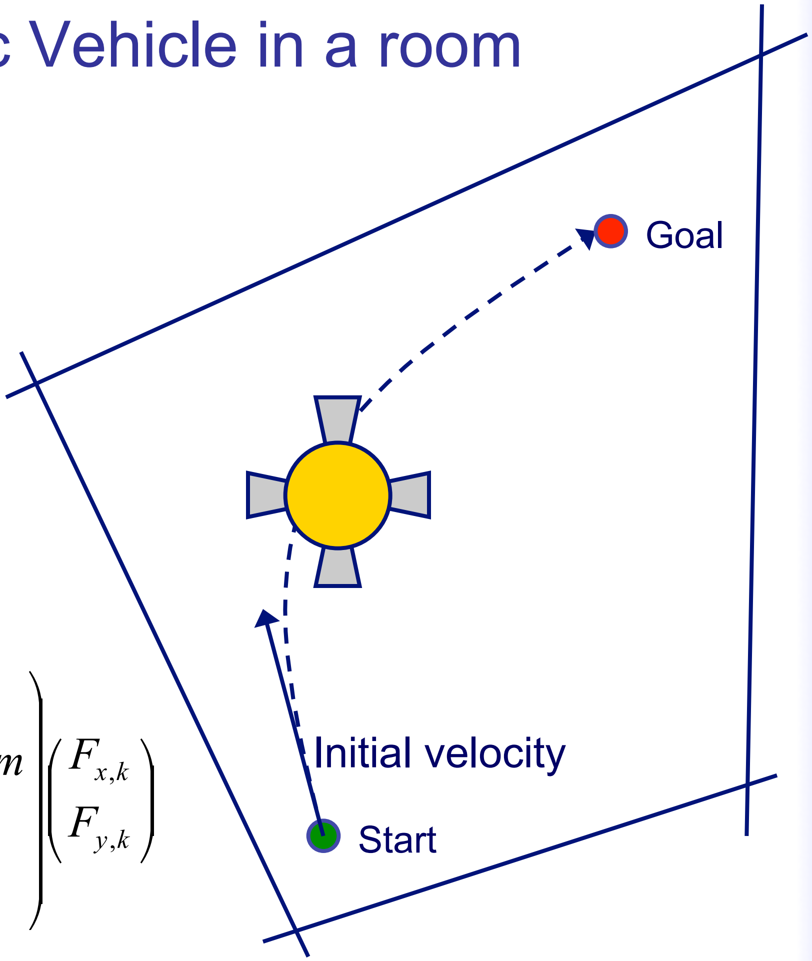
$$m\begin{pmatrix}\ddot{x}\\ \ddot{y}\end{pmatrix}=\begin{pmatrix}F_x\\ F_y\end{pmatrix}$$

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

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

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

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



***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

Example Constraints

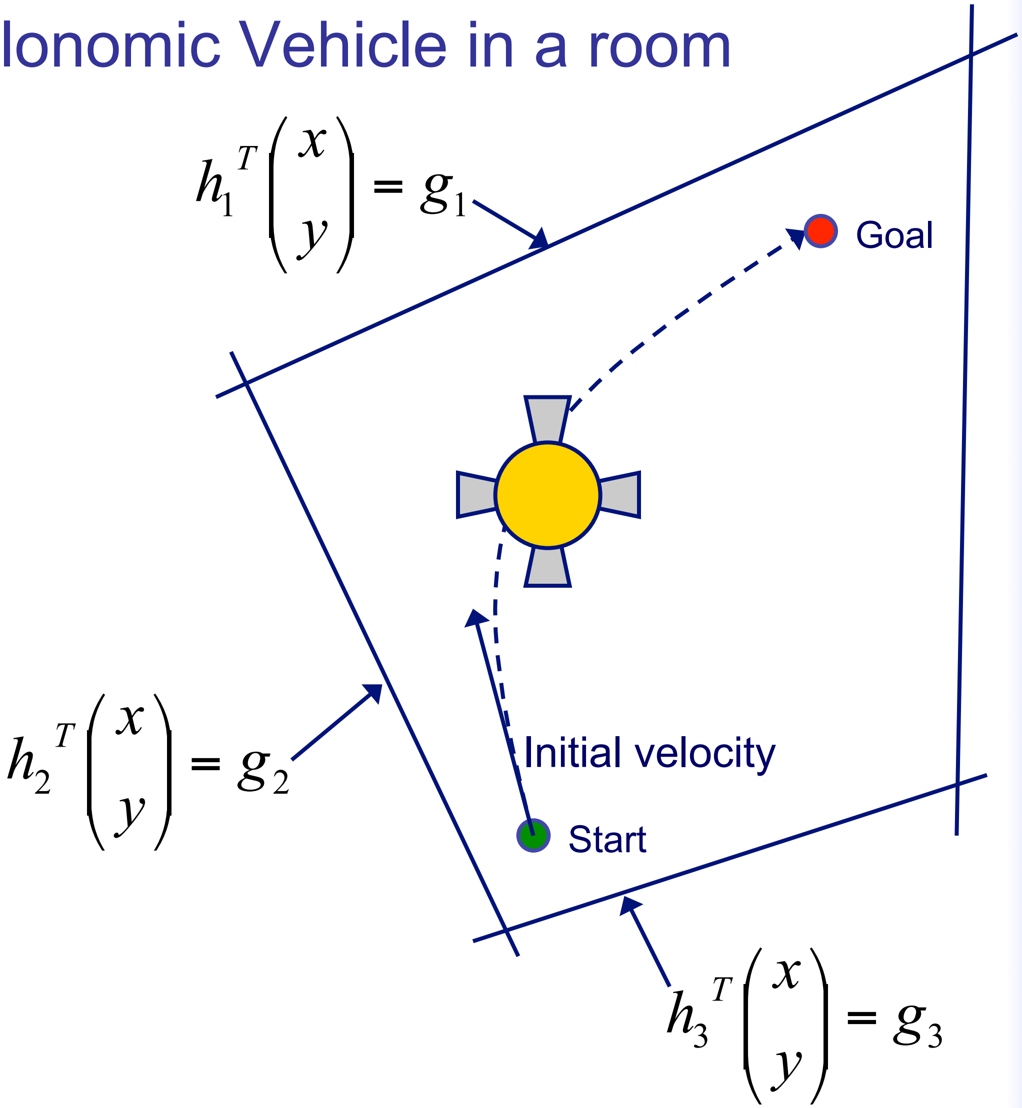
- 2-D Omni-dimensional Holonomic Vehicle in a room

Spatial constraints:
Vehicle must be in the room

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

or

$$\mathbf{H}\mathbf{x} \leq \mathbf{g}$$



$h_1^T \begin{pmatrix} x \\ y \end{pmatrix} = g_1$

$h_2^T \begin{pmatrix} x \\ y \end{pmatrix} = g_2$

$h_3^T \begin{pmatrix} x \\ y \end{pmatrix} = g_3$

Example Cost Function

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

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

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

$$\min |u| \quad \Longleftrightarrow \quad \begin{aligned} &\min u^+ + u^- \\ &u = u^+ - u^- \\ &u^+ \geq 0, u^- \geq 0, \end{aligned} \quad \textbf{or} \quad \begin{aligned} &\min v \\ &v \geq u, v \geq -u, \end{aligned}$$

Formulation of Receding Horizon Control

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

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

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

Spatial constraints

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

Initial position and velocity

It is not a good idea to fix N (time horizon)

$$\cancel{\mathbf{x}_{\enclose{circle}{N}} = \mathbf{x}_{\text{goal}}}$$

Goal position and velocity

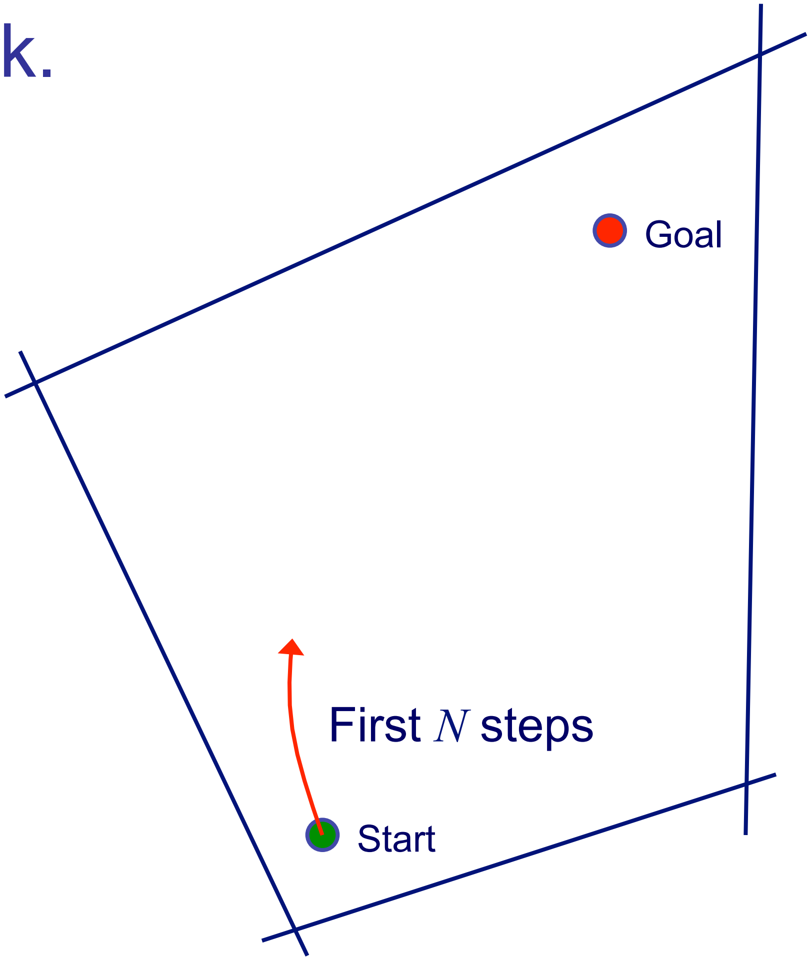
$$-\mathbf{u}_{\max} \leq \mathbf{u}_k \leq \mathbf{u}_{\max} \quad (k = 0,1,\cdots N-1)$$

Thrust limits

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

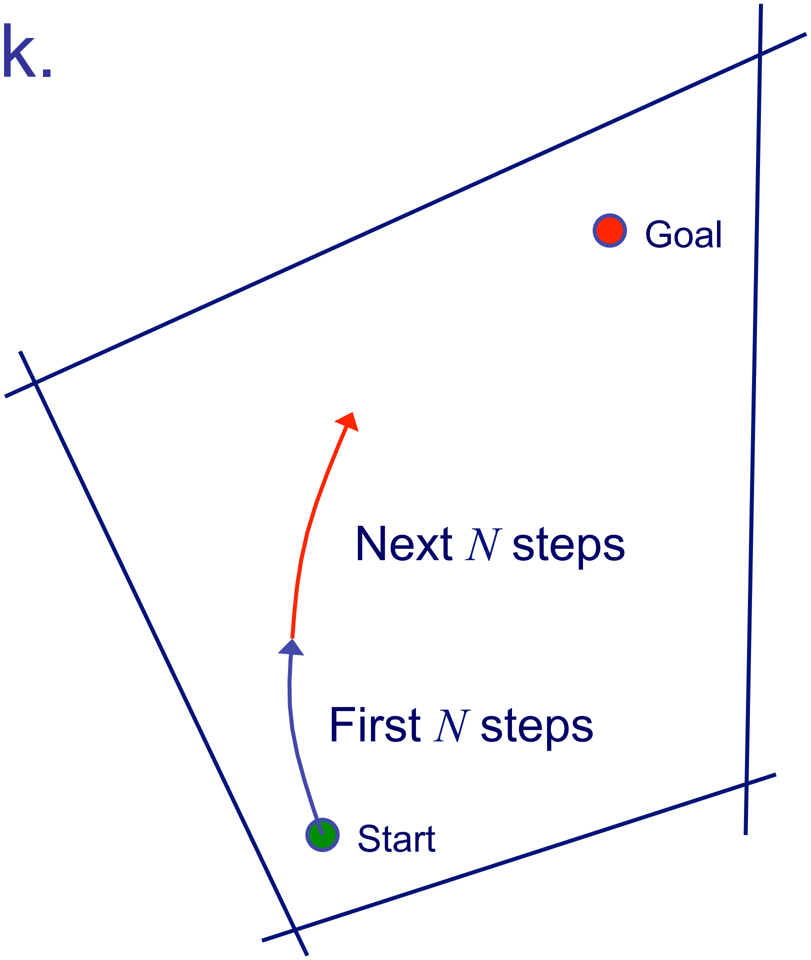
Receding Horizon Control

- Patchwork.



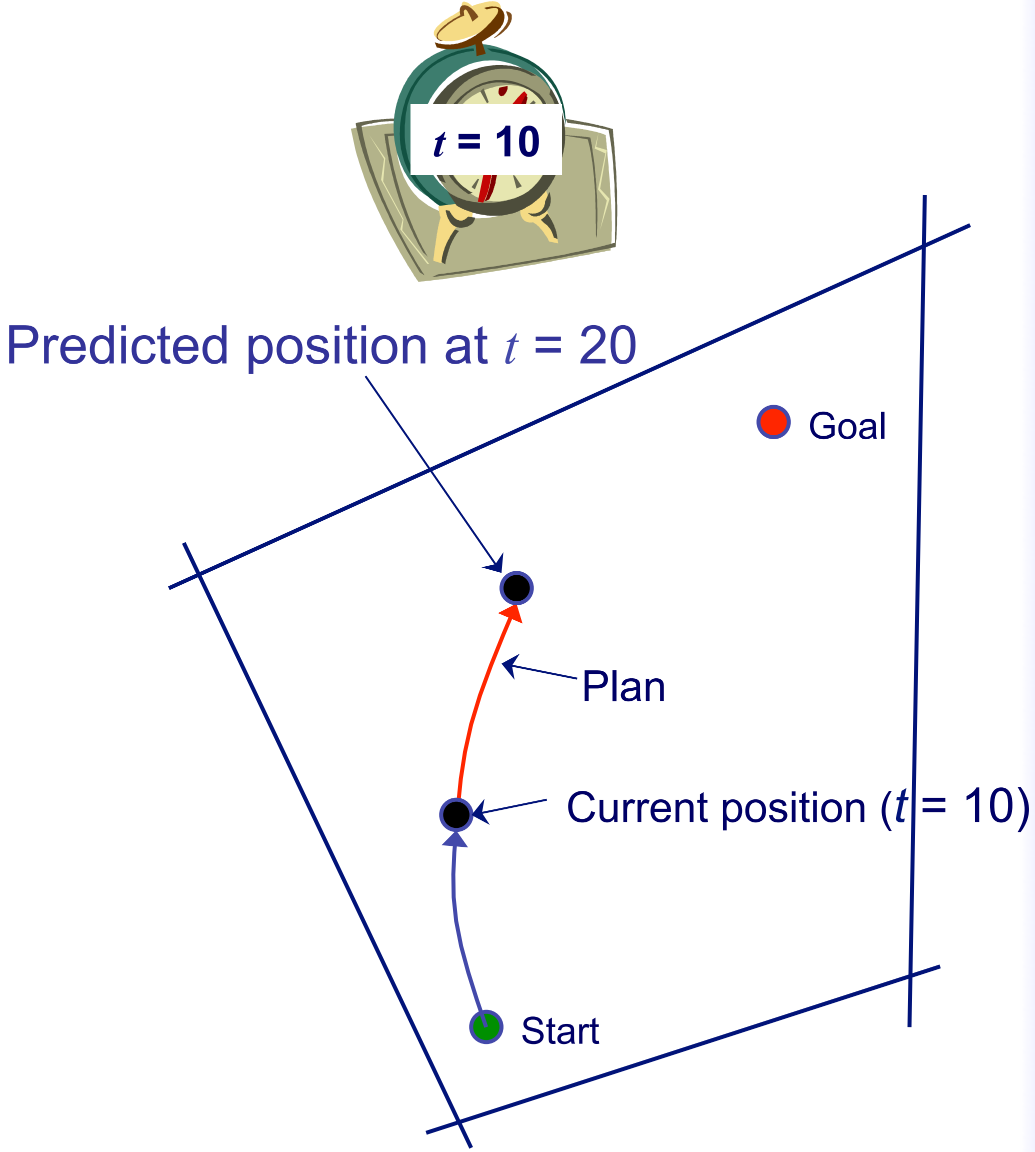
Receding Horizon Control

- Patchwork.



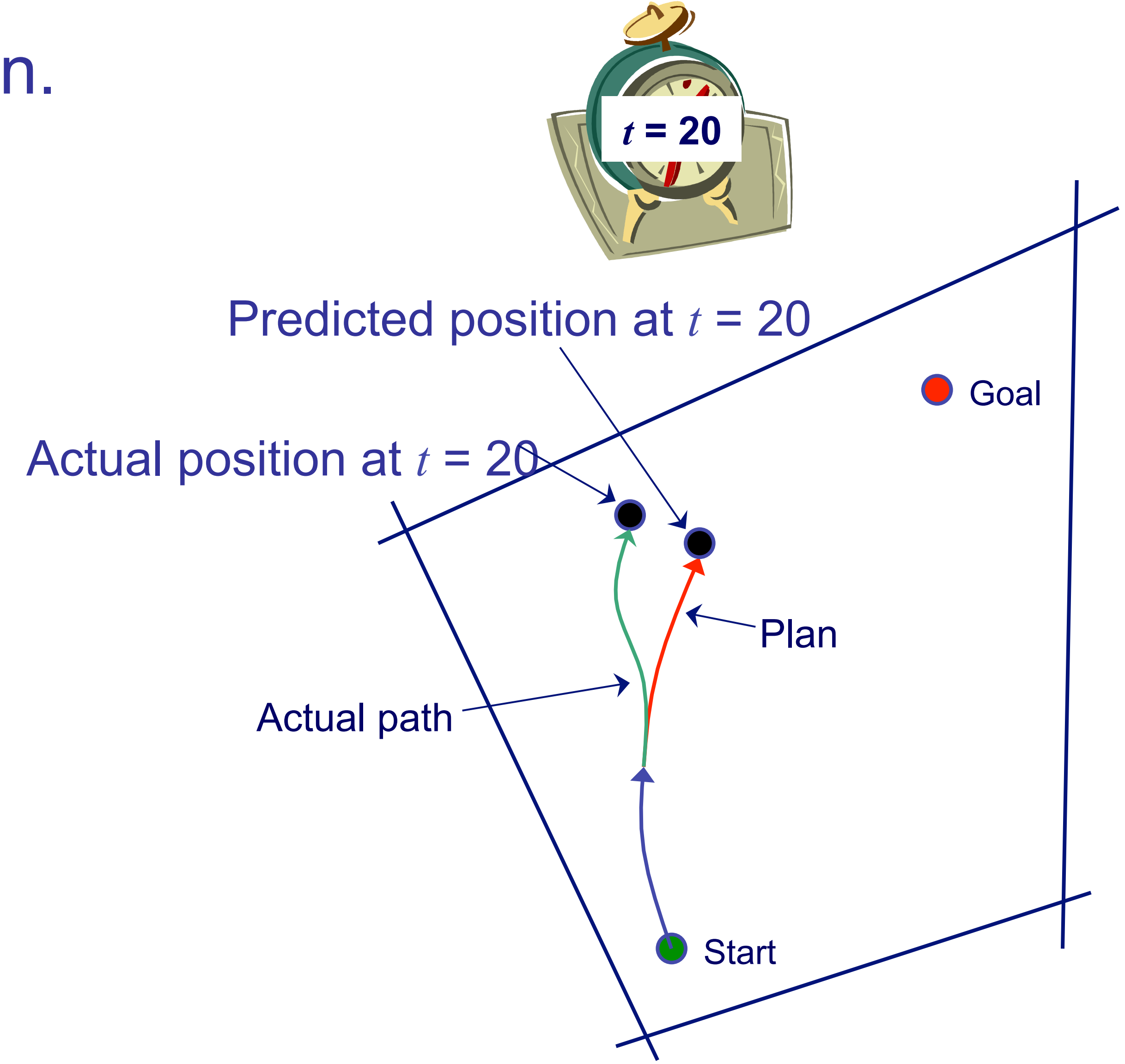
More on Receding Horizon

- 10 seconds later….



More on Receding Horizon

- World uncertain.



t = 20

Predicted position at t = 20

Goal

Actual position at t = 20

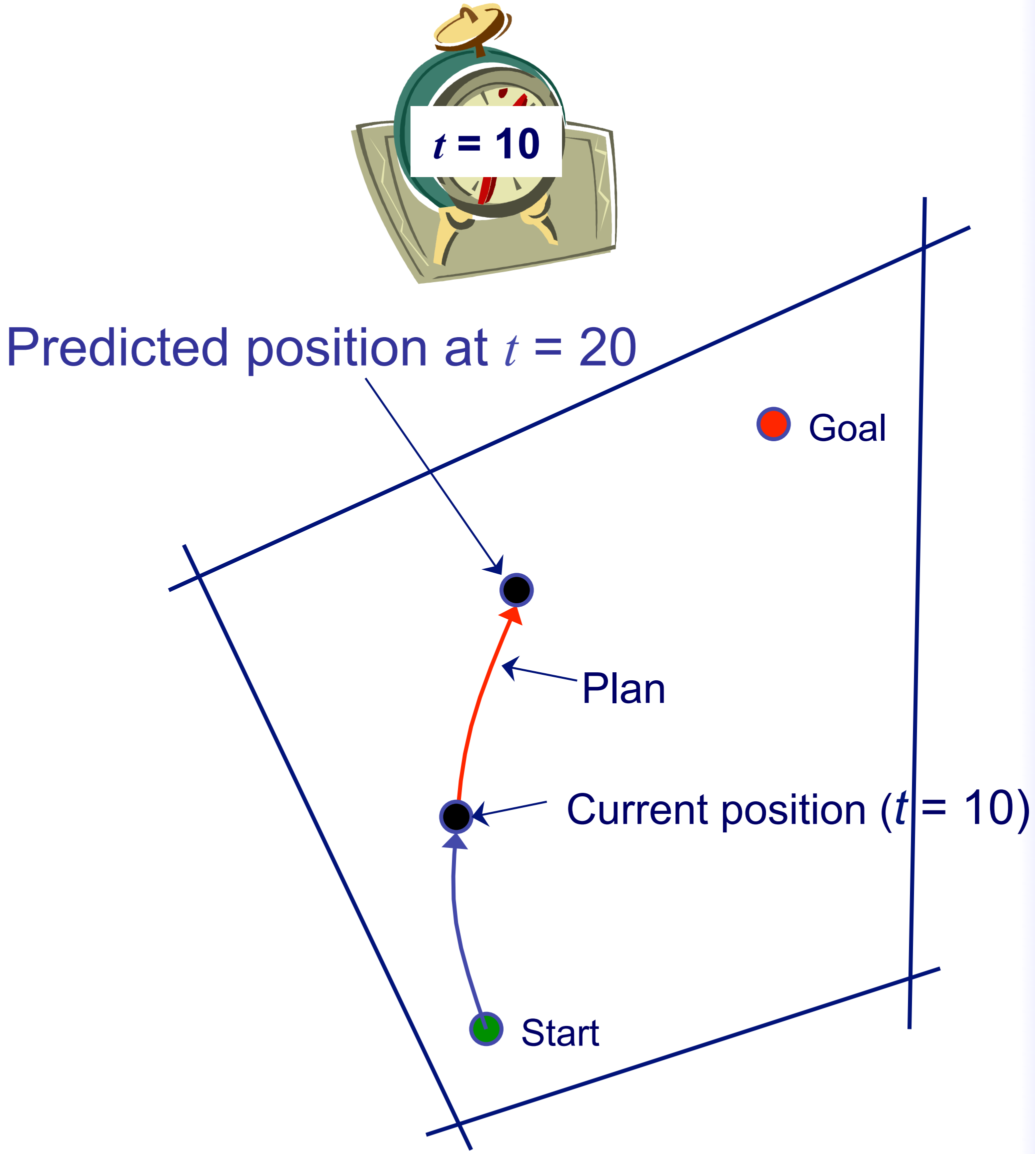
Plan

Actual path

Start

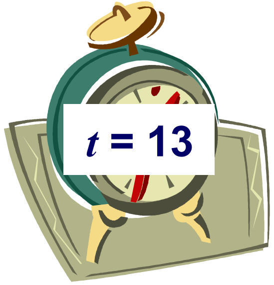
Execution Horizon < Planning Horizon

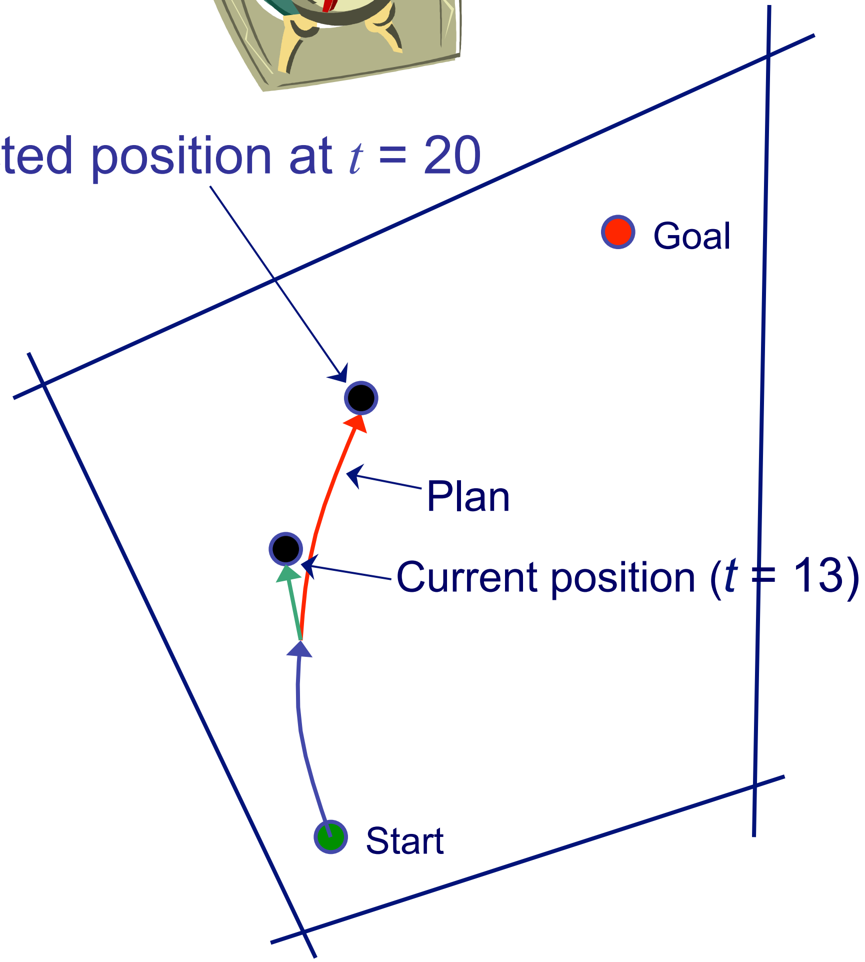
- 3 seconds later….



Execution Horizon < Planning Horizon

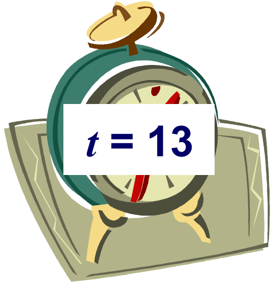
- 3 seconds later….
- Position a little bit off from the plan.

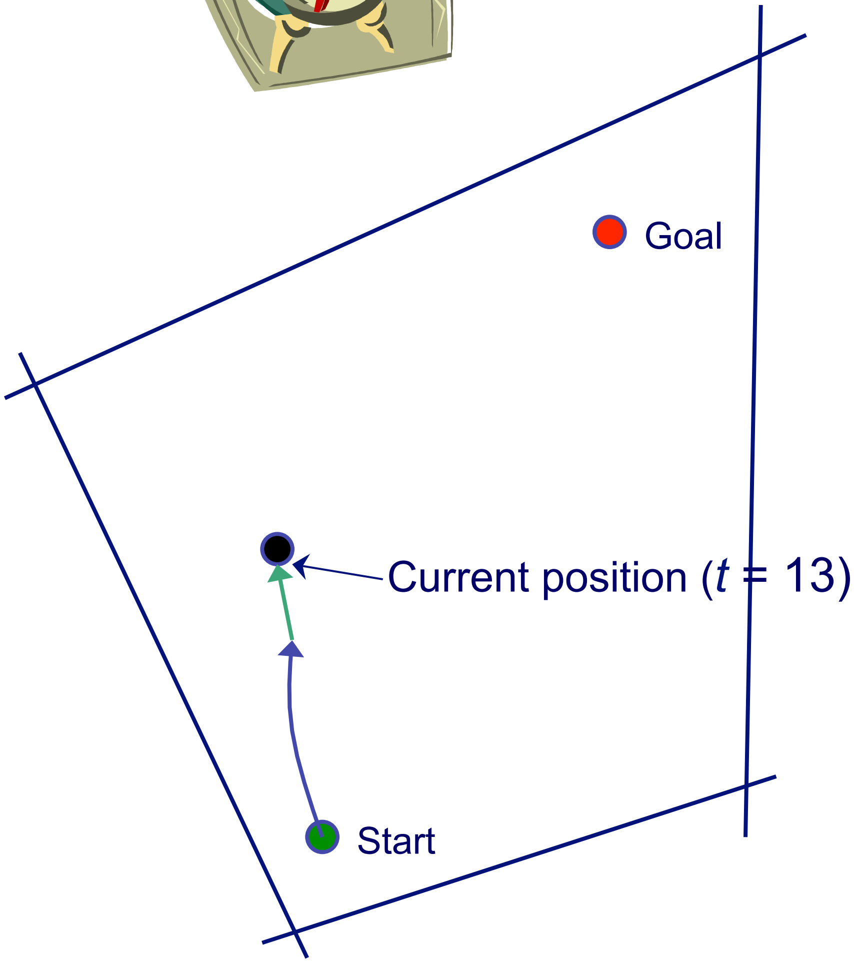




Execution Horizon < Planning Horizon

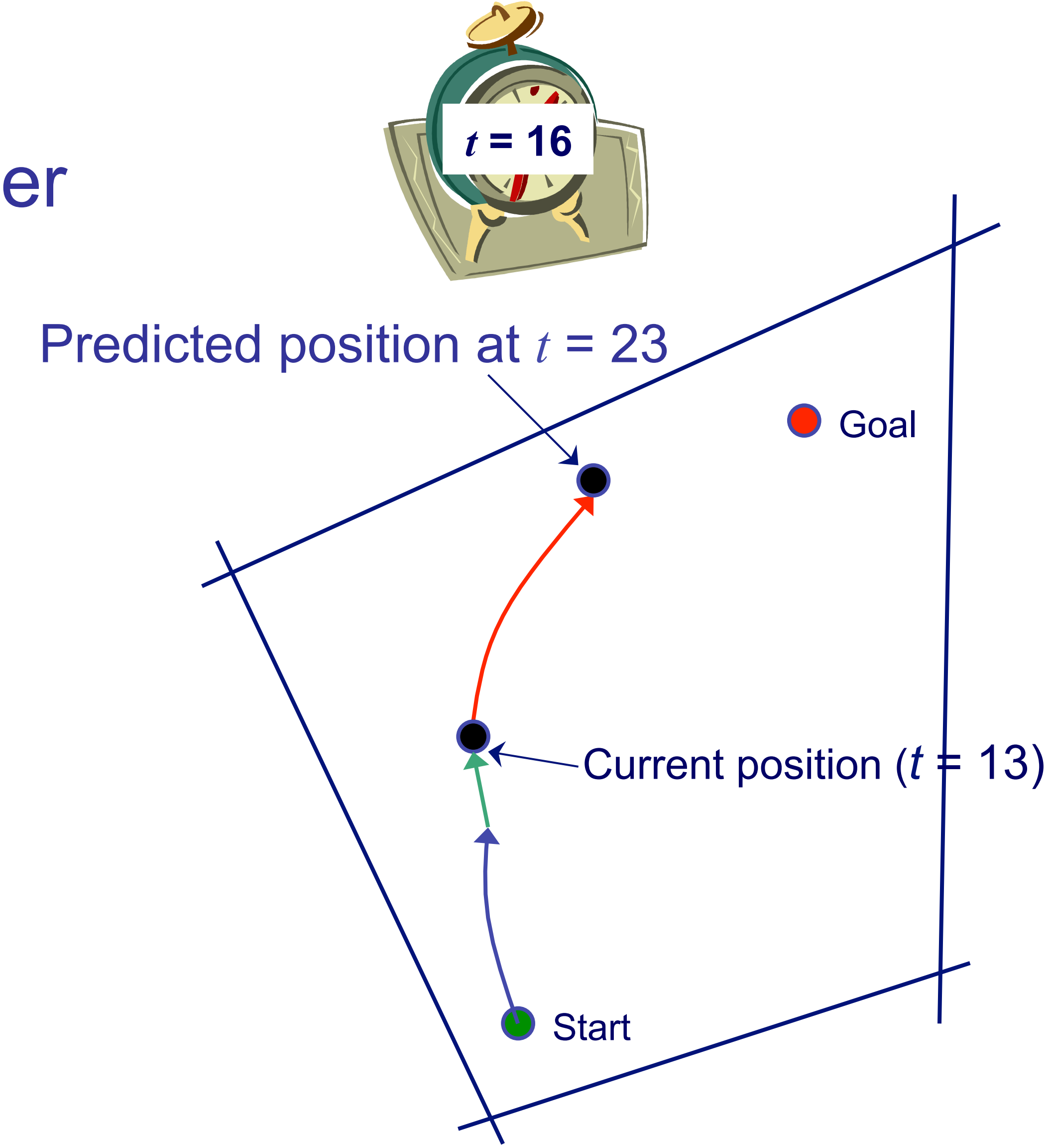
- Abandon the plan after t = 13.





Execution Horizon < Planning Horizon

- Abandon the plan after t = 13.
- Replan for another planning horizon.
- Repeat.



t = 16

Predicted position at t = 23

Goal

Current position (t = 13)

Start

Test bed: Connected Sustainable Home

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

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

Goal-directed Model-Predictive Control: Resident Goals

"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.

Goal-directed Model-Predictive Control: Resident Goals

Also, dry my clothes before morning.

Goal-directed Model-Predictive Control: Resident Goals

I need to use my car to drive to and from work, so make sure it is fully charged by morning.

Goal-directed Model-Predictive Control: Resident Goals

It's acceptable if my clothes aren't ready by morning or if the house is a couple degrees too cold, but my car absolutely needs to be ready to use before I leave for work."

Goal-directed Model-Predictive Control: Resident Goals

"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. Also, dry my clothes before morning. I need to use my car to drive to and from work, so make sure it is fully charged by morning. It's acceptable if my clothes aren't ready by morning or if the house is a couple degrees too cold, but my car absolutely needs to be ready to use before I leave for work."

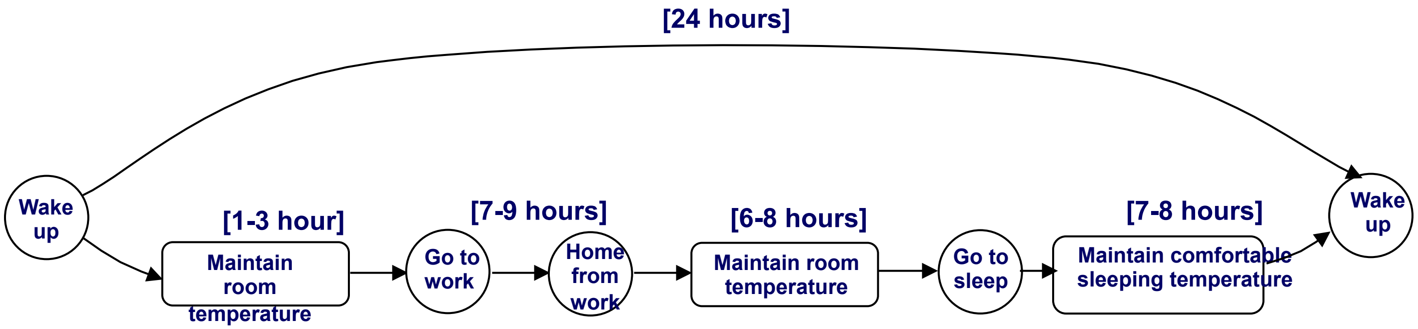
Flexibility Available to Control

- When activities are performed.
- When to charge/discharge batteries.
- Which activities to shed (when supply is low).

Encoding: Qualitative State Plan (QSP)

Model-based Embedded & Robotic Systems

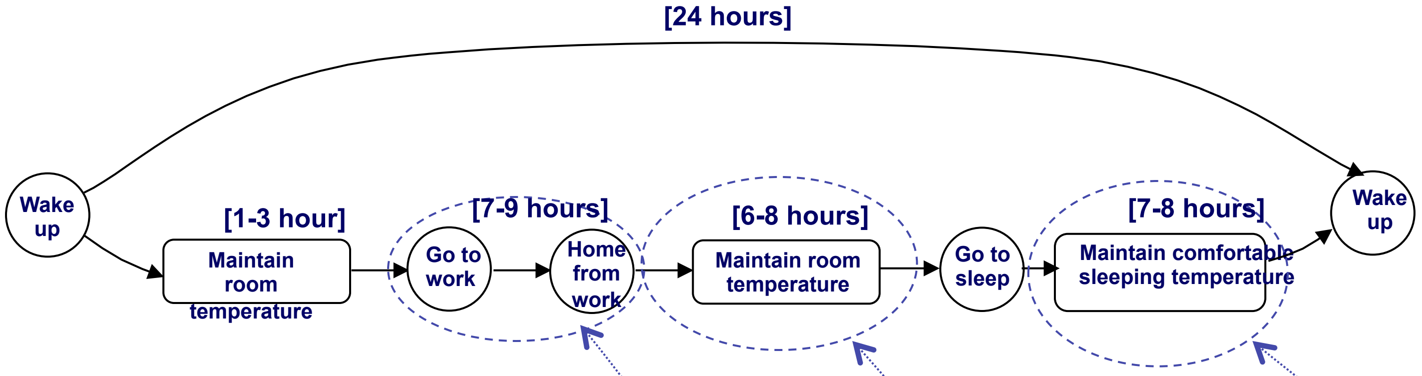
Sulu [Leaute & Williams, AAAI05]



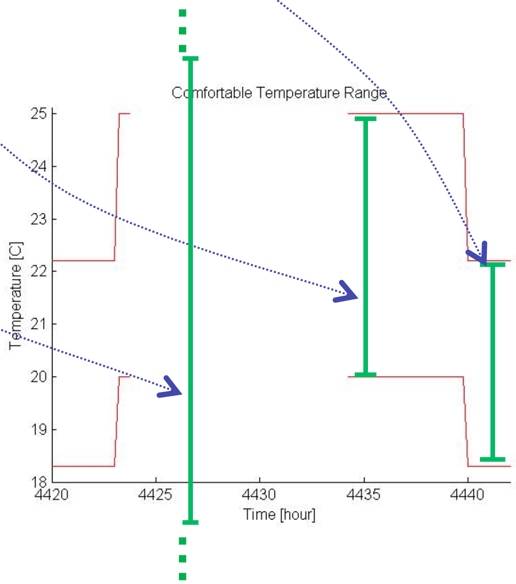
"Maintain room temperature after waking up until I go to work. No temperature constraints while I'm at work, but when I get home, maintain room temperature until I go to sleep. Maintain a comfortable sleeping temperature while I sleep."

Encoding: 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."



Encode the Qualitative State Plan and Dynamics within a Model-Predictive Controller

Sulu [Leaute & Williams, AAAI05]

Cost

$$\min_{\mathbf{U}} J(\mathbf{X},\mathbf{U}) + H(x_T)$$

$$\boldsymbol{s.t.}$$

Dynamics

(Discrete time)

$$\forall_{0 \le t \le T-1} x_{t+1} = Ax_t + Bu_t$$

Constraints

$$\bigwedge_{t=0}^{T} \bigwedge_{i=0}^{N} \bigvee_{j=0}^{M} h_t^{iT} x_t \le g_t^{ij}$$

State

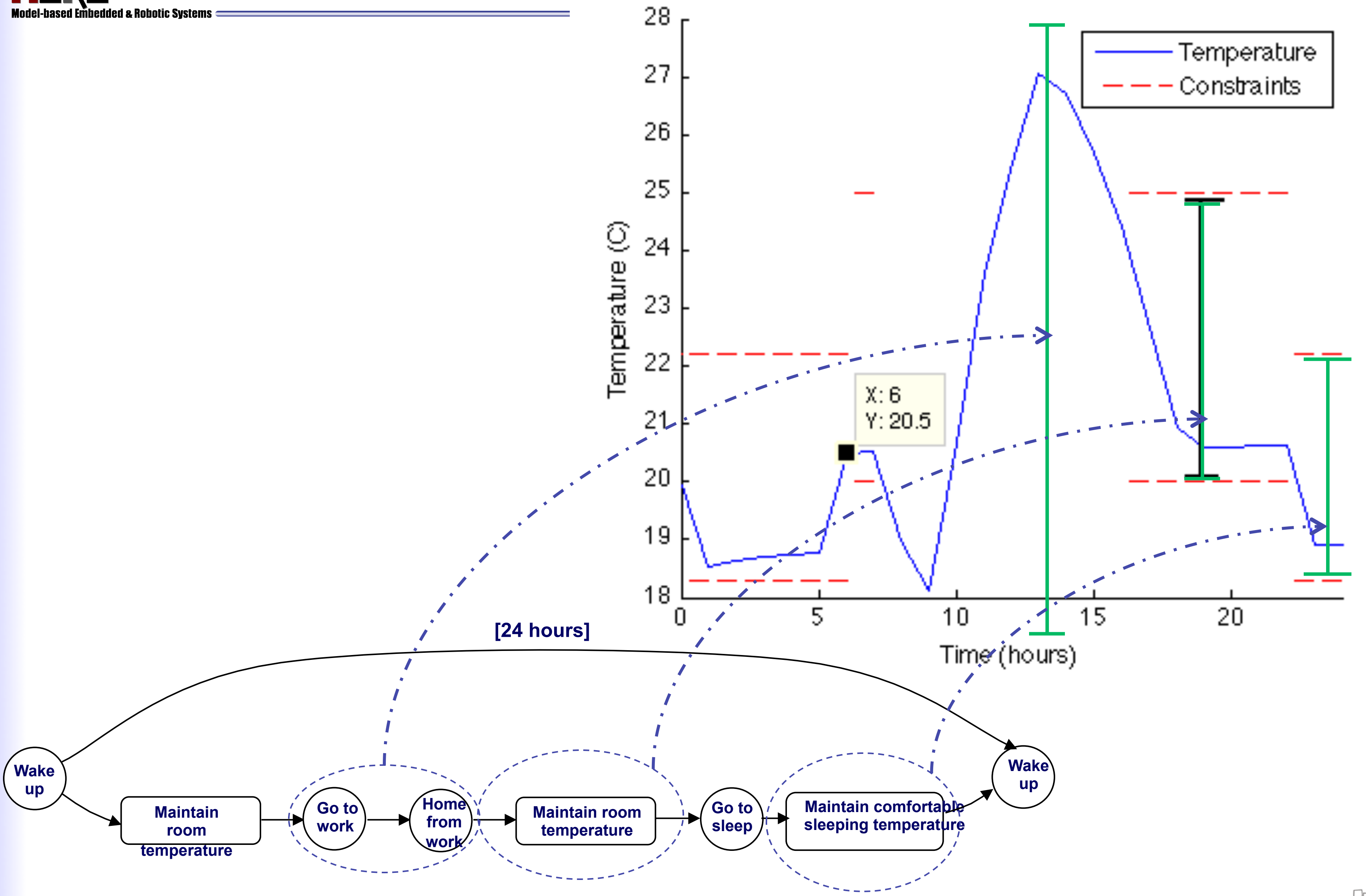
$$\mathbf{X} = \left[x_0 \cdots x_t\right]^T$$

Control

$$\mathbf{U} = \left[u_0 \cdots u_{t-1}\right]^T$$

Mixed Integer and Logic

(p)Sulu Results



[24 hours]

Wake up

Maintain room temperature

Go to work

Home from work

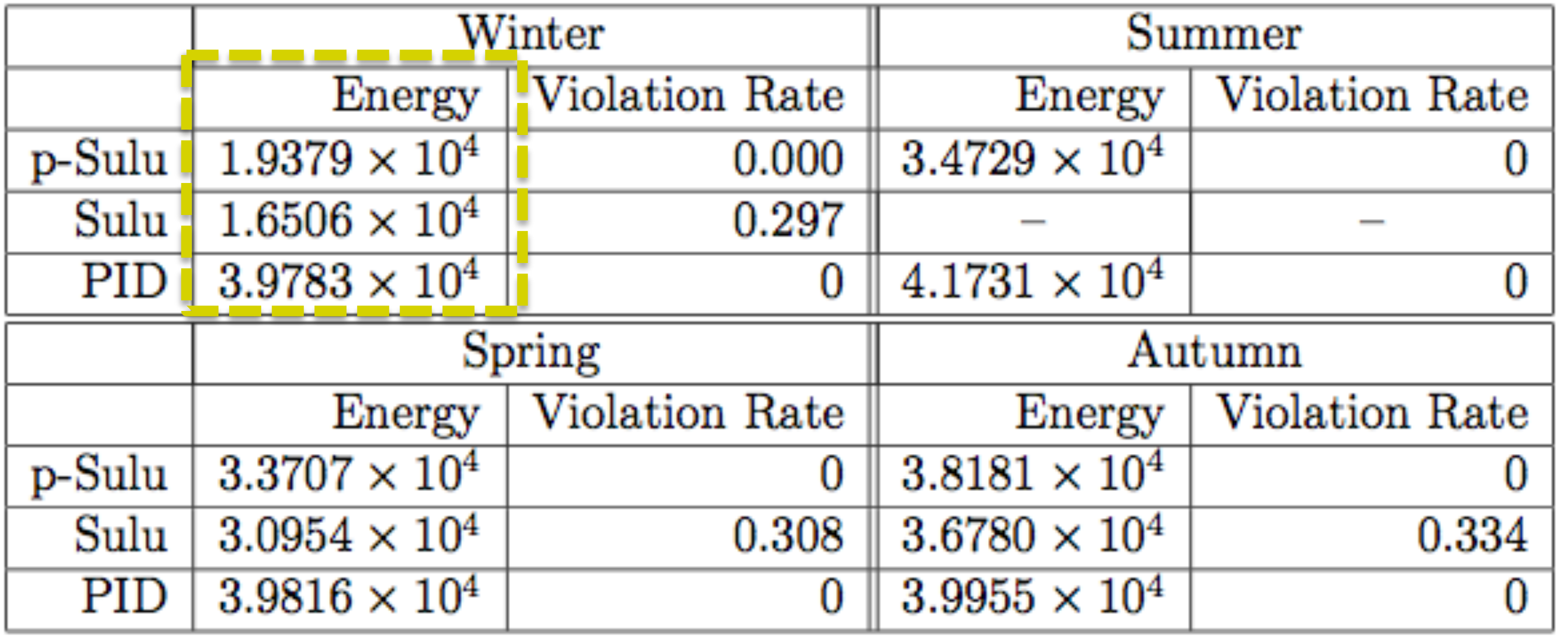
Maintain room temperature

Go to sleep

Maintain comfortable sleeping temperature

Wake up

Energy Savings: Optimal Control

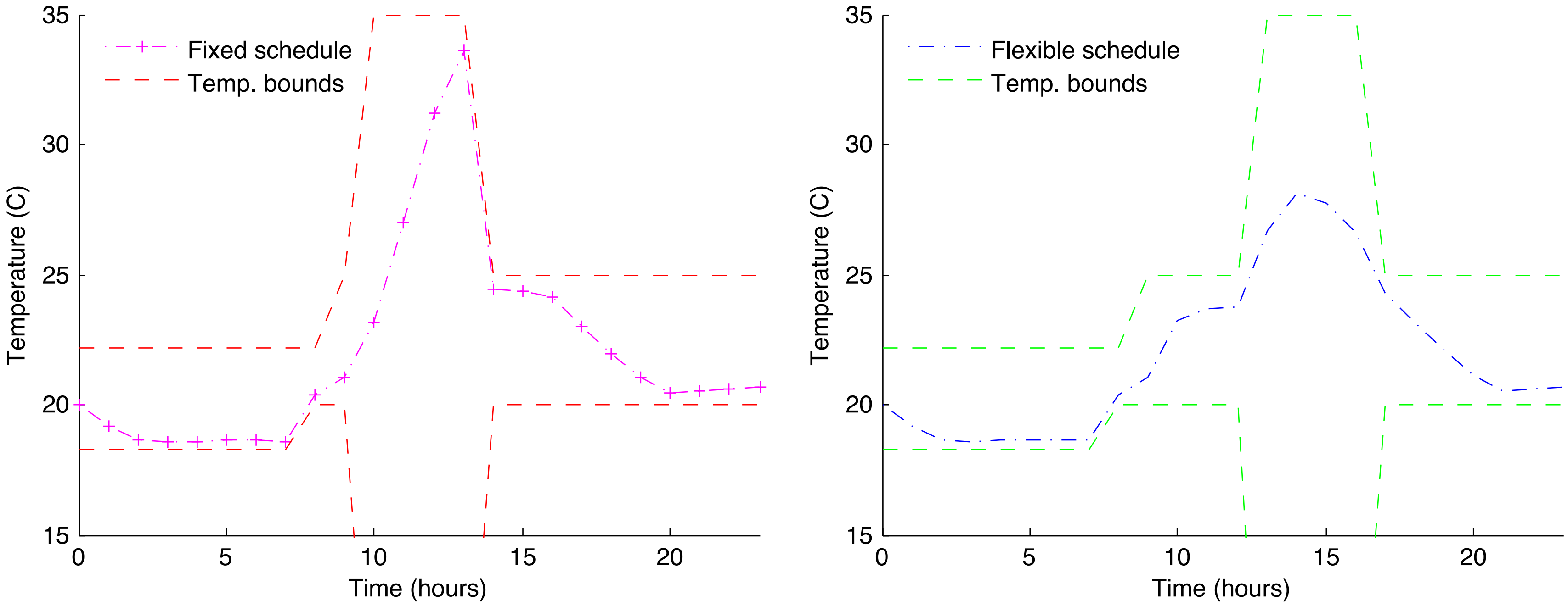


	Winter		Summer	
	Energy	Violation Rate	Energy	Violation Rate
p-Sulu	1.9379×10^4	0.000	3.4729×10^4	0
Sulu	1.6506×10^4	0.297	–	–
PID	3.9783×10^4	0	4.1731×10^4	0

	Spring		Autumn	
	Energy	Violation Rate	Energy	Violation Rate
p-Sulu	3.3707×10^4	0	3.8181×10^4	0
Sulu	3.0954×10^4	0.308	3.6780×10^4	0.334
PID	3.9816×10^4	0	3.9955×10^4	0

- 42.8% savings in winter over PID.
- 15.3%, 16.8%, and 4.4% in spring, summer, autumn.

Additional Savings Due to Flexibility

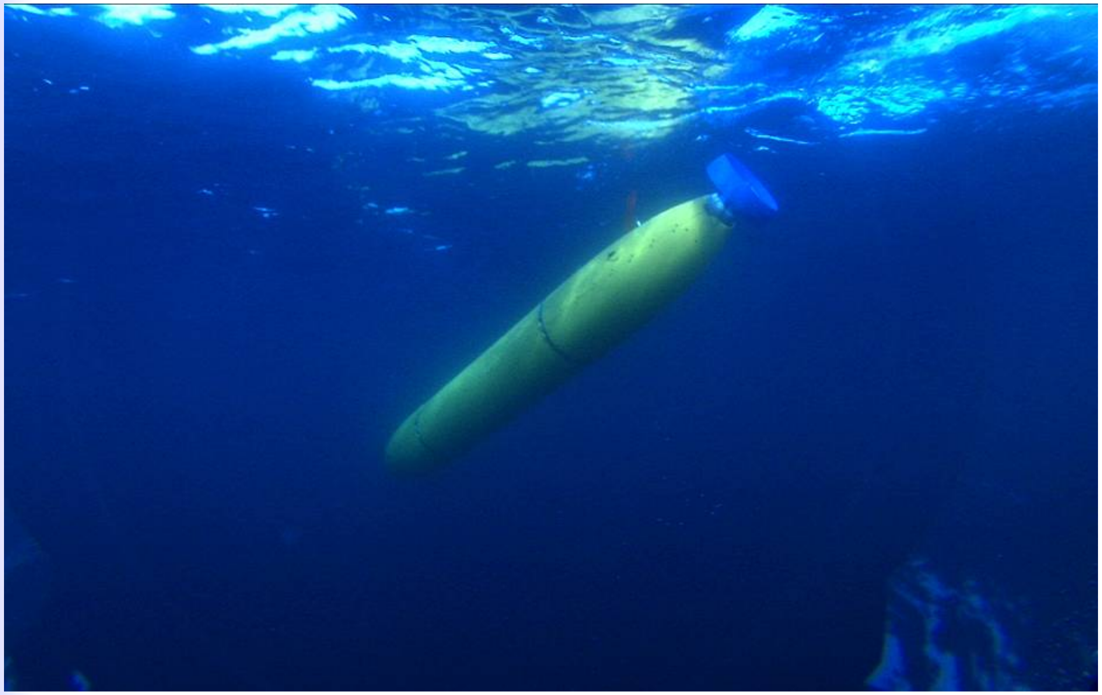


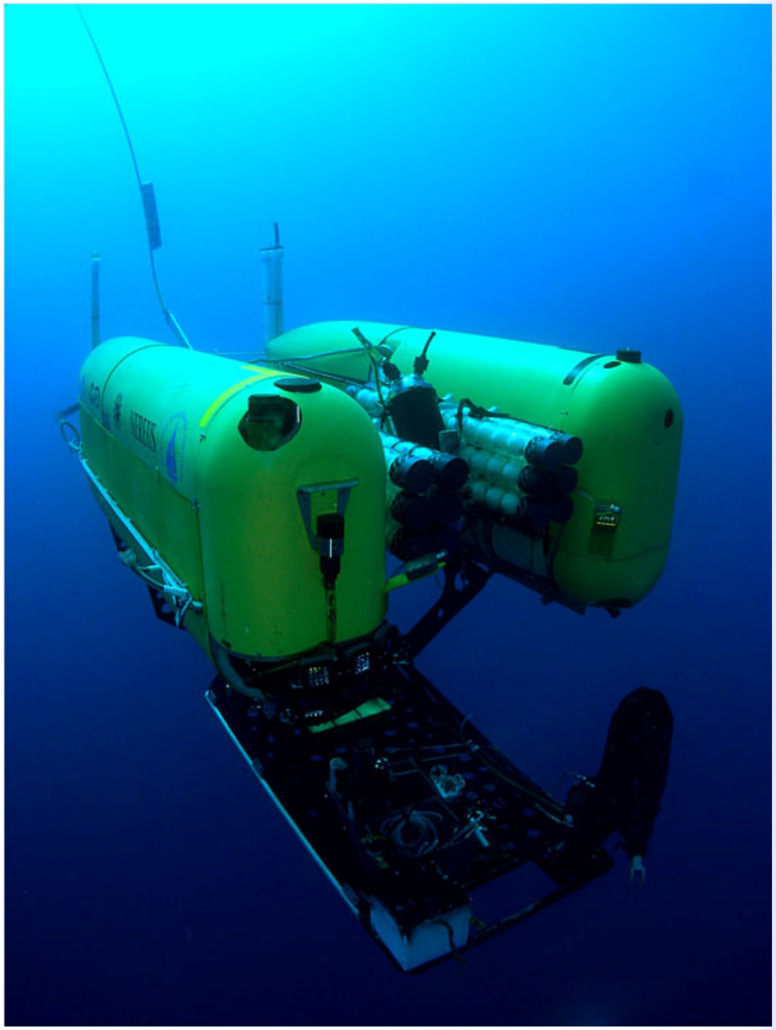
- 10.4%, 1.6%, 1.6%, and 0.7% in the winter, spring, summer, and autumn.

Outline

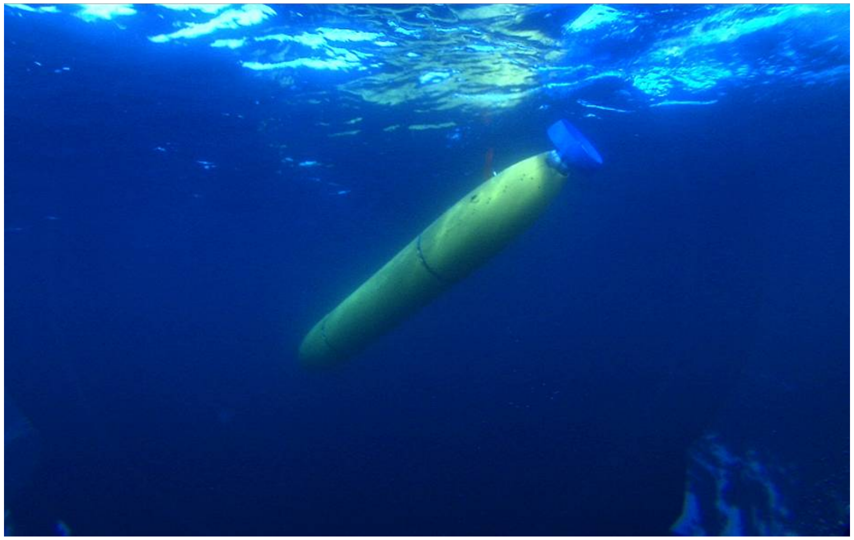
- Goal-directed, Model-Predictive Control
- Stochastic Optimization
- Iterative Risk Allocation
- Optimal Risk Allocation
- Appendix: Multi-agent Risk Allocation

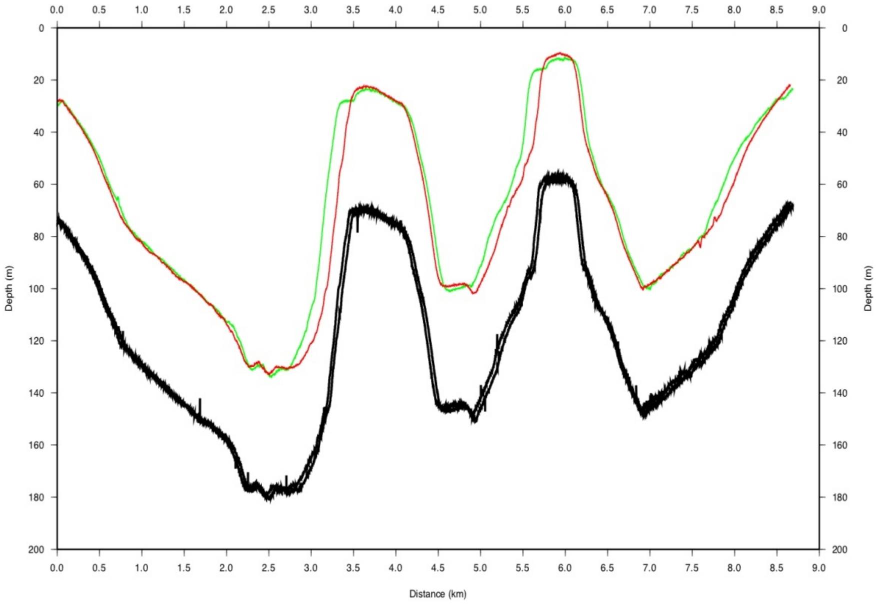
The Danger of Ignoring Uncertainty

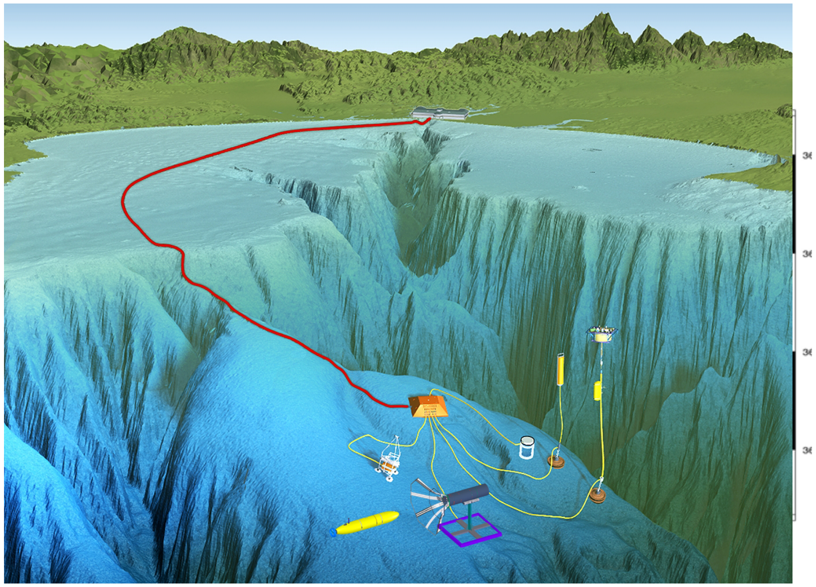


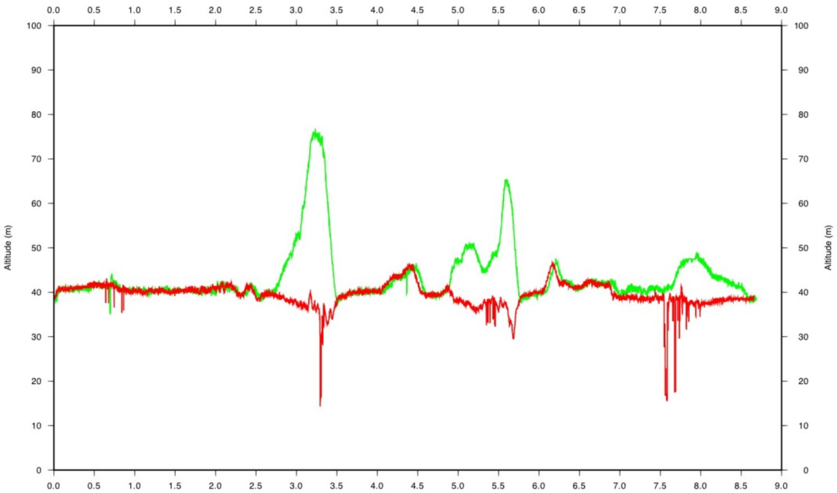


Problem: Managing Risk within Mission-Guidelines





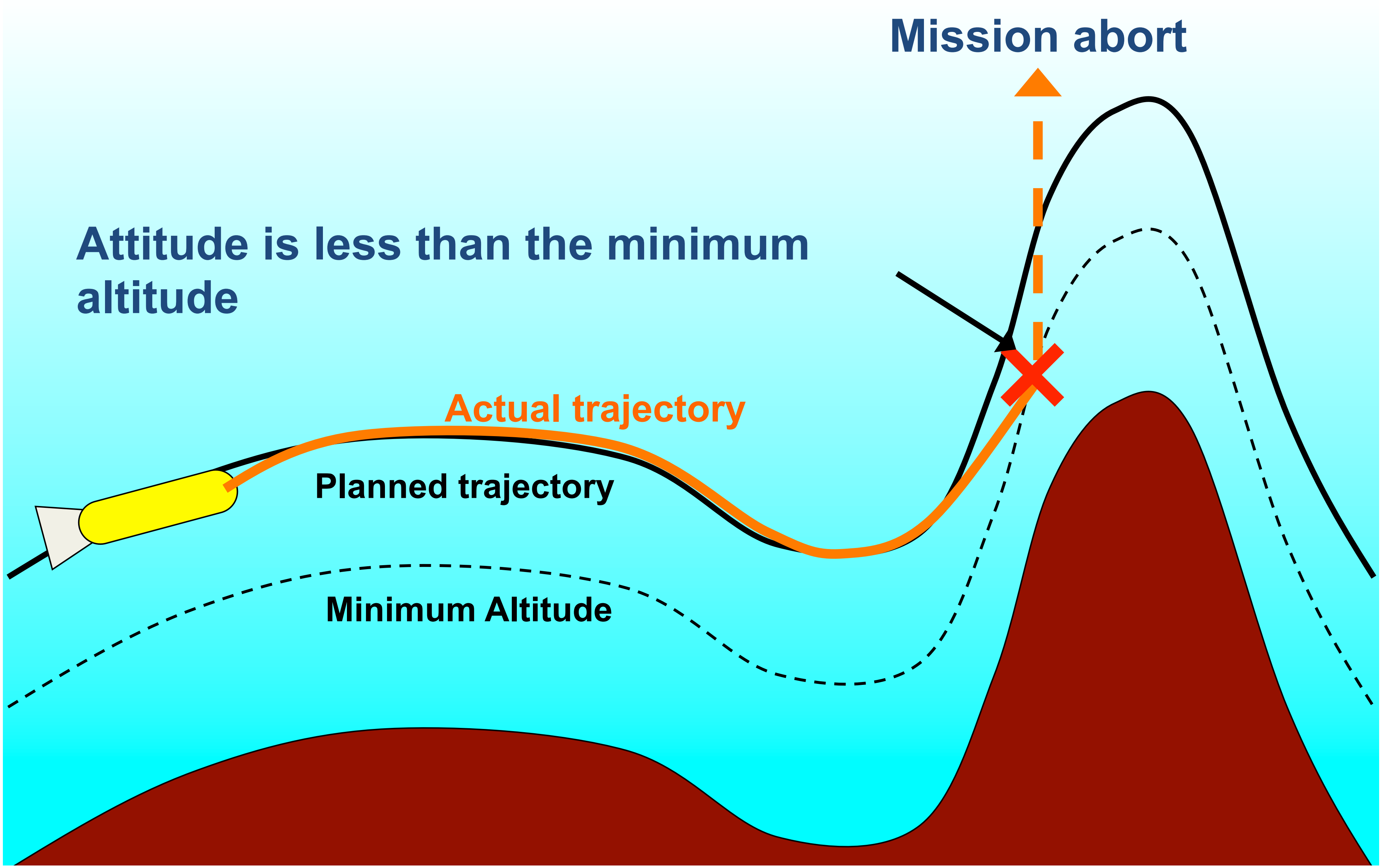





>$1M
1 25002 74135 0

Model-based Embedded & Robotic Systems

Issue: Frequent Mission Aborts



Robust Model Predictive Control

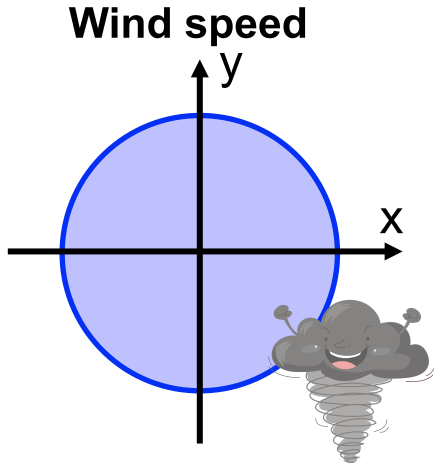
- Receding horizon planners react to uncertainty ***after*** something goes wrong.
- Can't we take precautionary actions ***before*** something goes wrong?

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

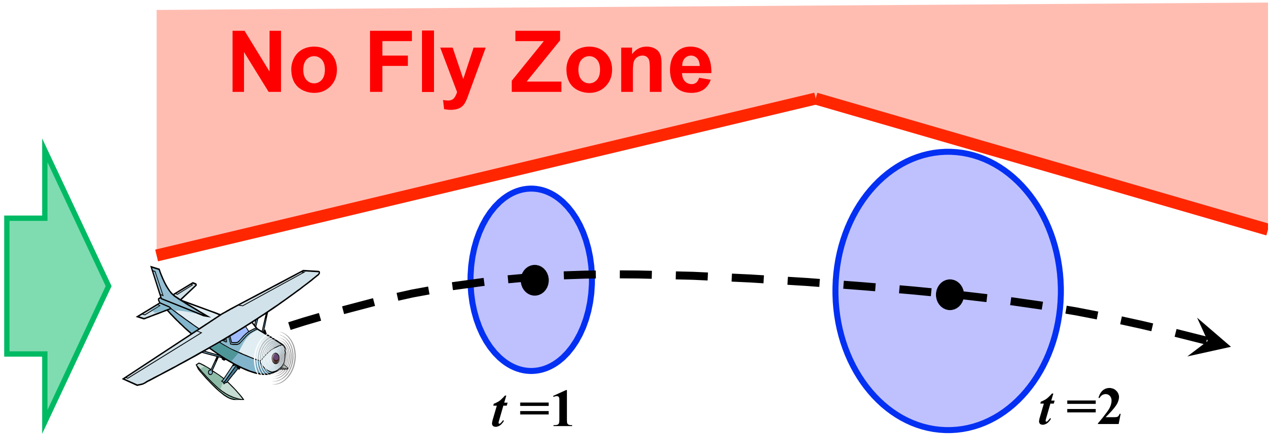
Model-based Embedded & Robotic Systems

Robust versus Chance Constrained

Robust Predictive Control

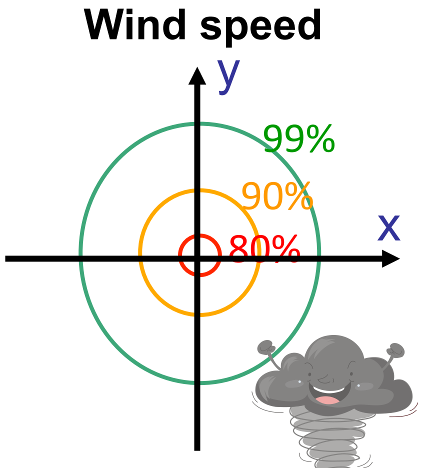


Assumes **bounds** on uncertainty.

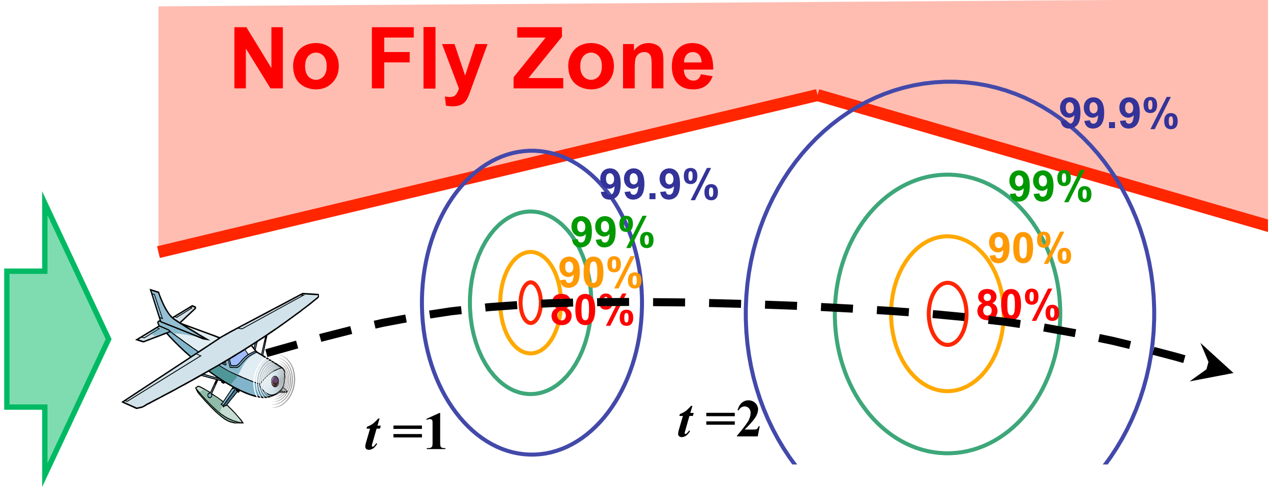


- Predicted position has **bounded** uncertainty.
- Problem: Find a control sequence that satisfies the constraints for **all realizations of uncertainty.**

Chance-constrained Predictive Control



Assumes a **probability distribution** that characterizes uncertainty.



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

Incorporating Uncertainty

- Deterministic discrete-time LTI model

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

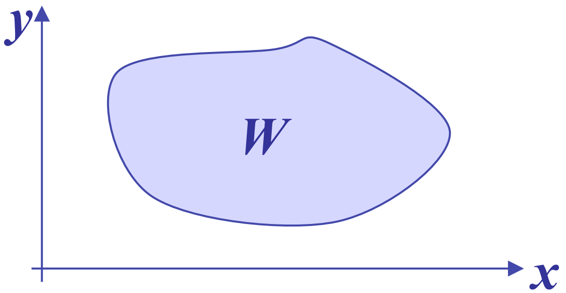
- Additive uncertainty

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

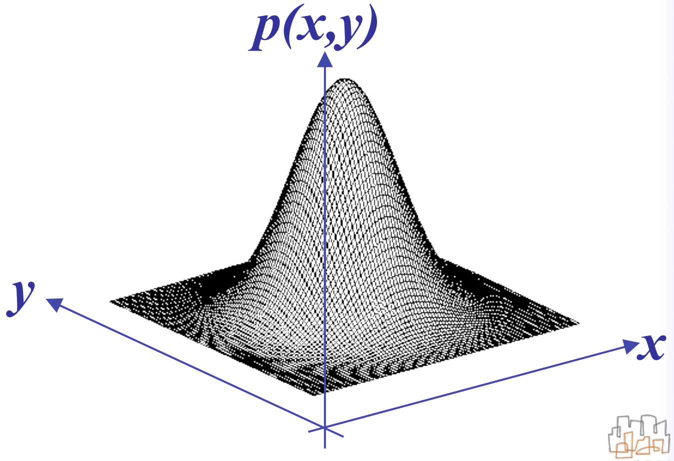
- Multiplicative uncertainty

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

$$w_t \in W$$



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



What to Minimize? (*Bounded* Uncertainty)

- Minimize the worst case cost

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

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

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

- Minimize nominal cost

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

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

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

What to Minimize? (*Stochastic* Uncertainty)

- Utilitarian approach

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

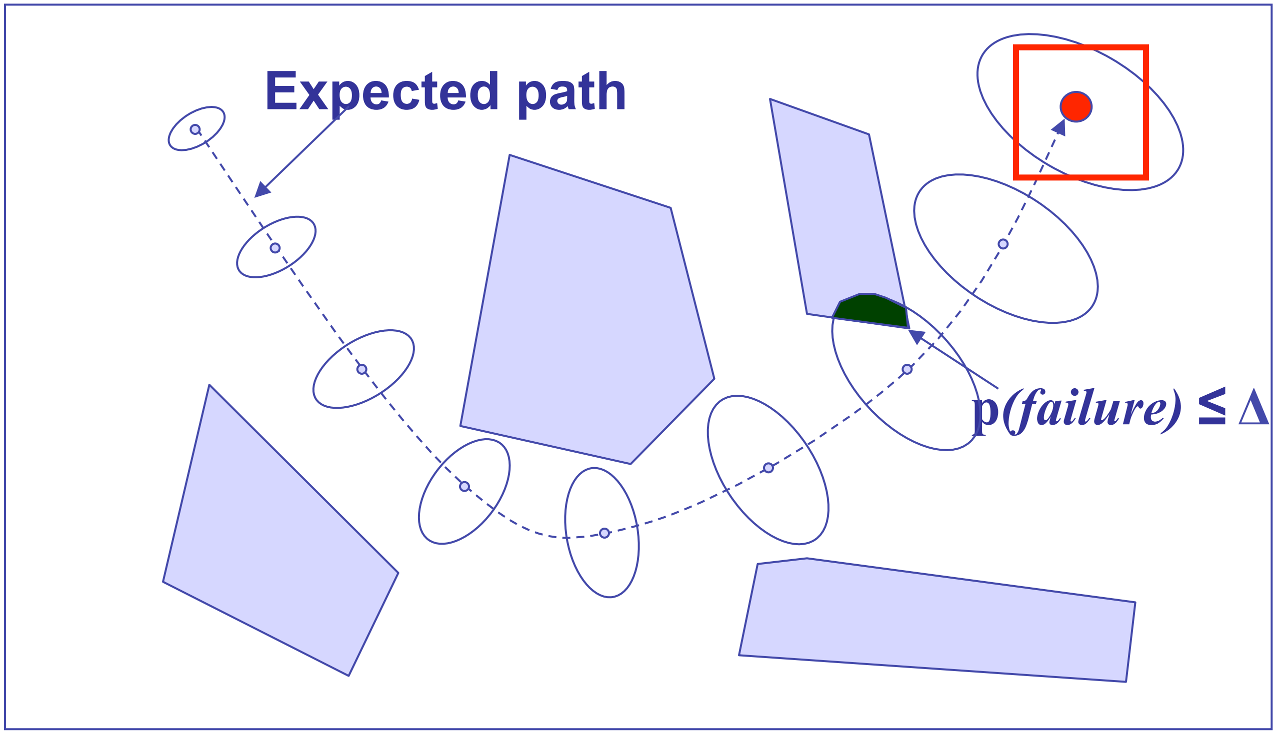
- **Chance constrained optimization**

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

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

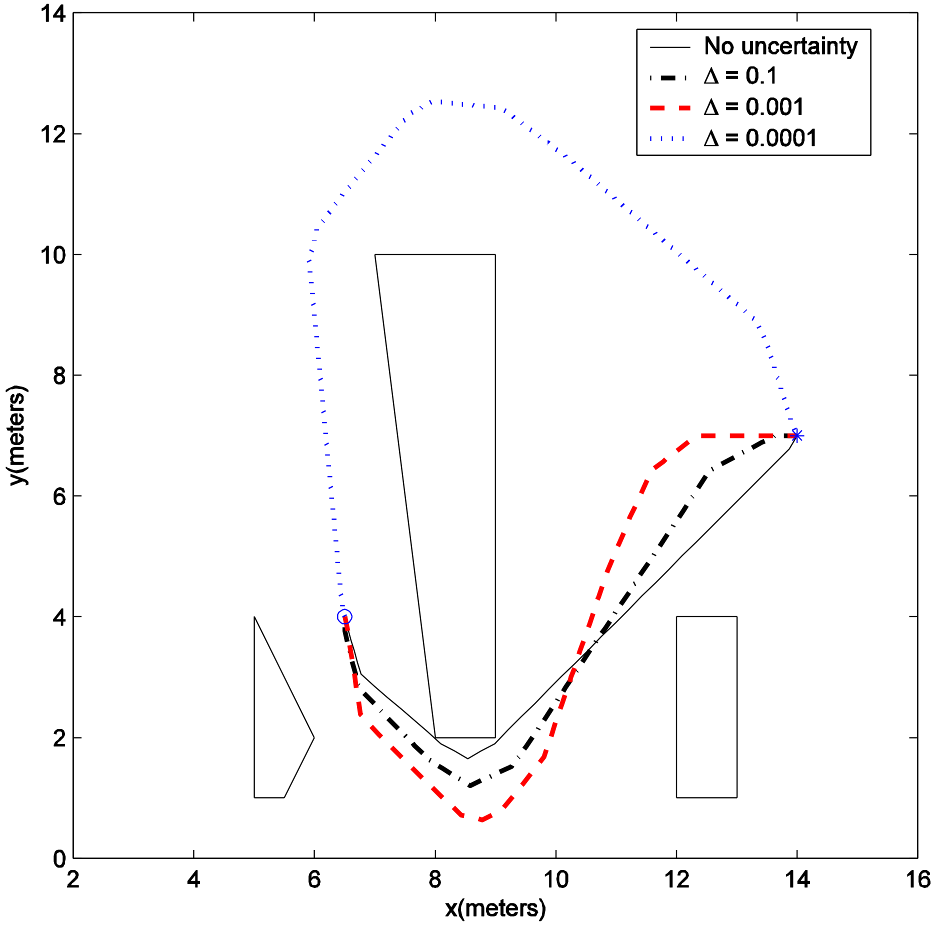
Chanced Constrained, Robust Path Planning

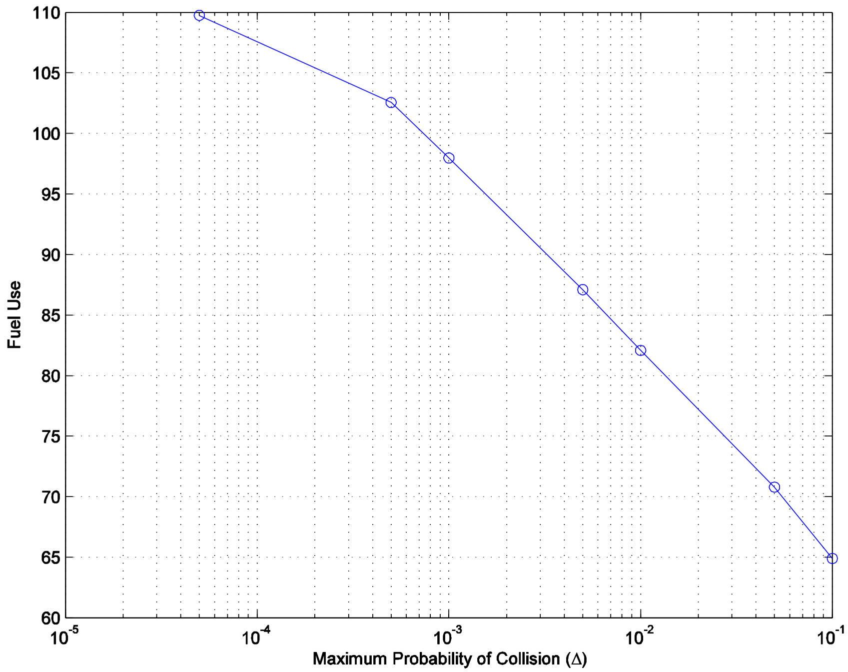
- “Plan optimal path to goal such that p*(failure)* ≤ Δ.”



Risk – Performance Tradeoff

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





Method: Uniform Risk Allocation [Blackmore, PhD]

RMPC with Chance Constraints

- MPC

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

$$s.t.$$

Dynamics

(Discrete time)

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

Constraints

$$\bigwedge_{t=0}^{T} \bigwedge_{i=0}^{N} h_t^{iT} x_t \le g_t^i$$

RMPC with Chance Constraints

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

$$s.t.$$

Stochastic dynamics $$\forall_{0 \le t \le T-1} \; x_{t+1} = Ax_t + Bu_t + w_t$$

Constraints $$\bigwedge_{t=0}^{T} \bigwedge_{i=0}^{N} h_t^{iT} x_t \le g_t^i$$

RMPC with Chance Constraints

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

$$s.t.$$

Stochastic dynamics

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

Gaussian distribution

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

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

Constraints

$$\bigwedge_{t=0}^{T} \bigwedge_{i=0}^{N} h_t^{iT} x_t \le g_t^i$$

RMPC with Chance Constraints

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

$$s.t.$$

Stochastic dynamics

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

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

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

Constraints

$$\bigwedge_{t=0}^{T} \bigwedge_{i=0}^{N} h_t^{iT} x_t \le g_t^i$$

RMPC with Chance Constraints

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

$$s.t.$$

Stochastic dynamics

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

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

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

Upper bound on the probability of failure = *Risk bound.*

Chance constraint

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

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

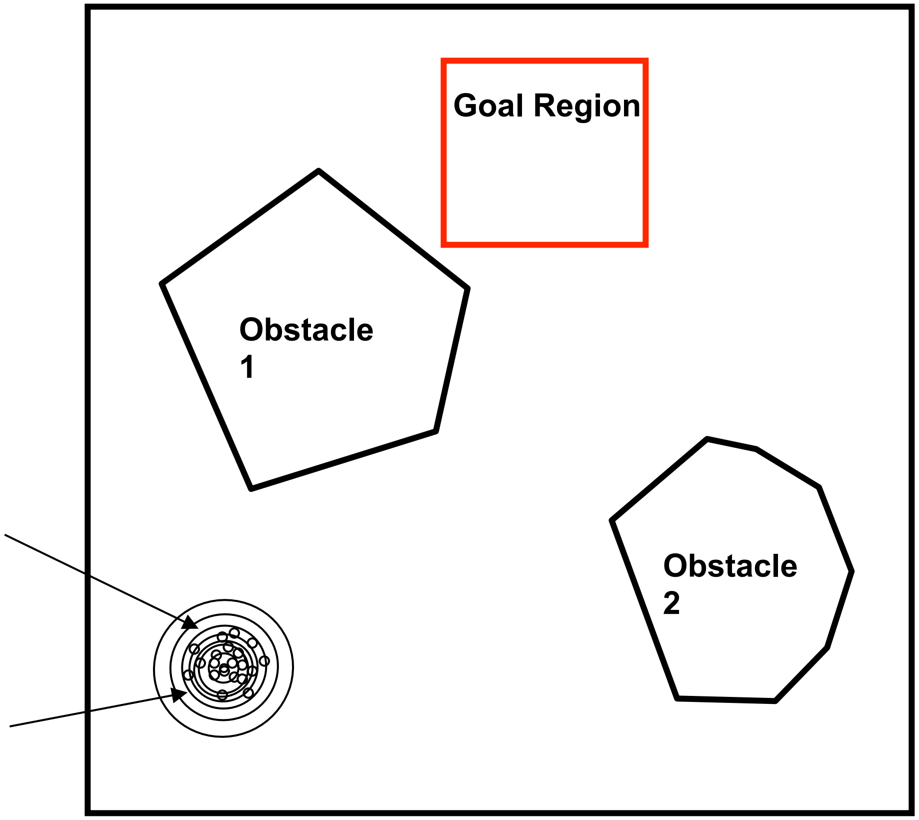
Solution Methods for Chance-Constrained Problems

- Sampling based methods
  - Scenario-based (see Warren Powell's tutorial).
    - 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.

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



Initial state distribution.

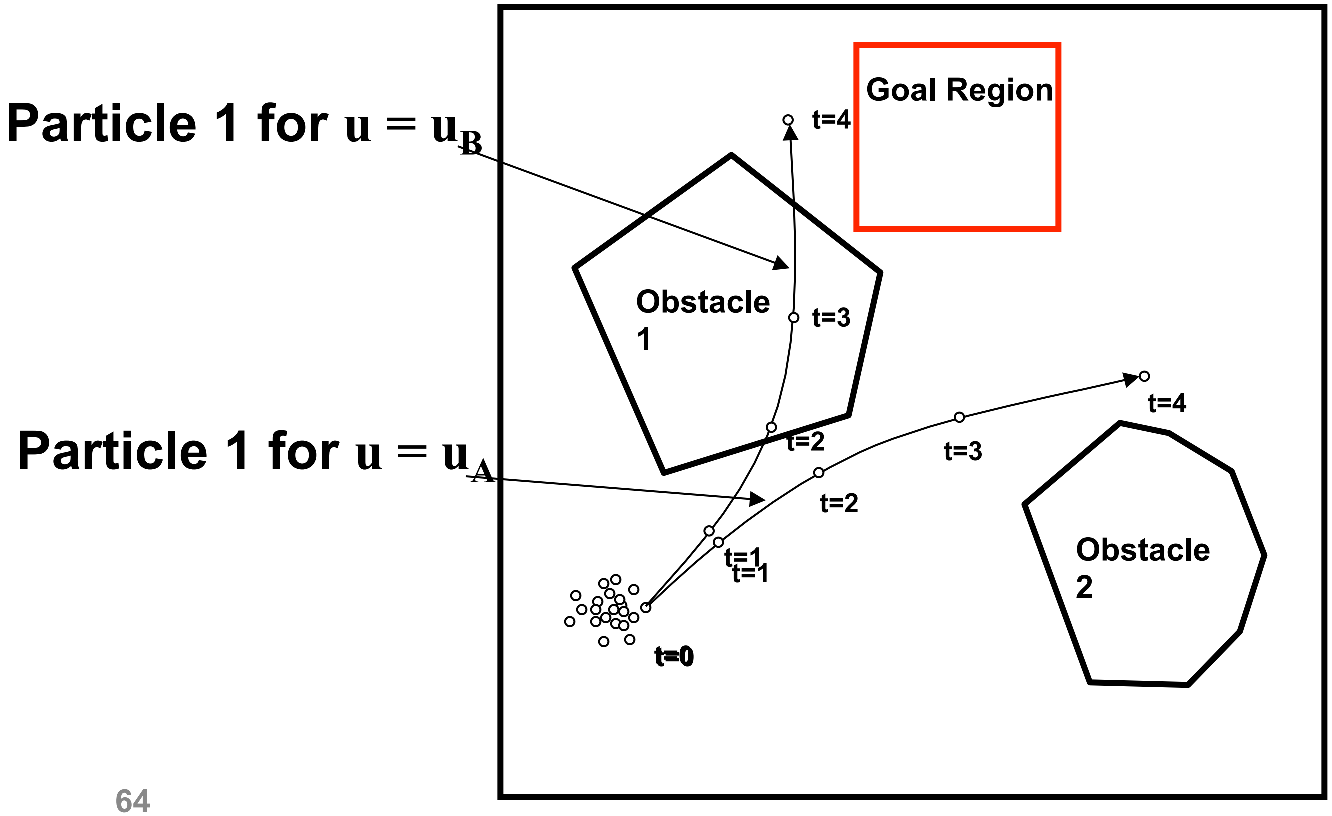
Particles approximating initial state distribution.

Particle Control

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

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

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

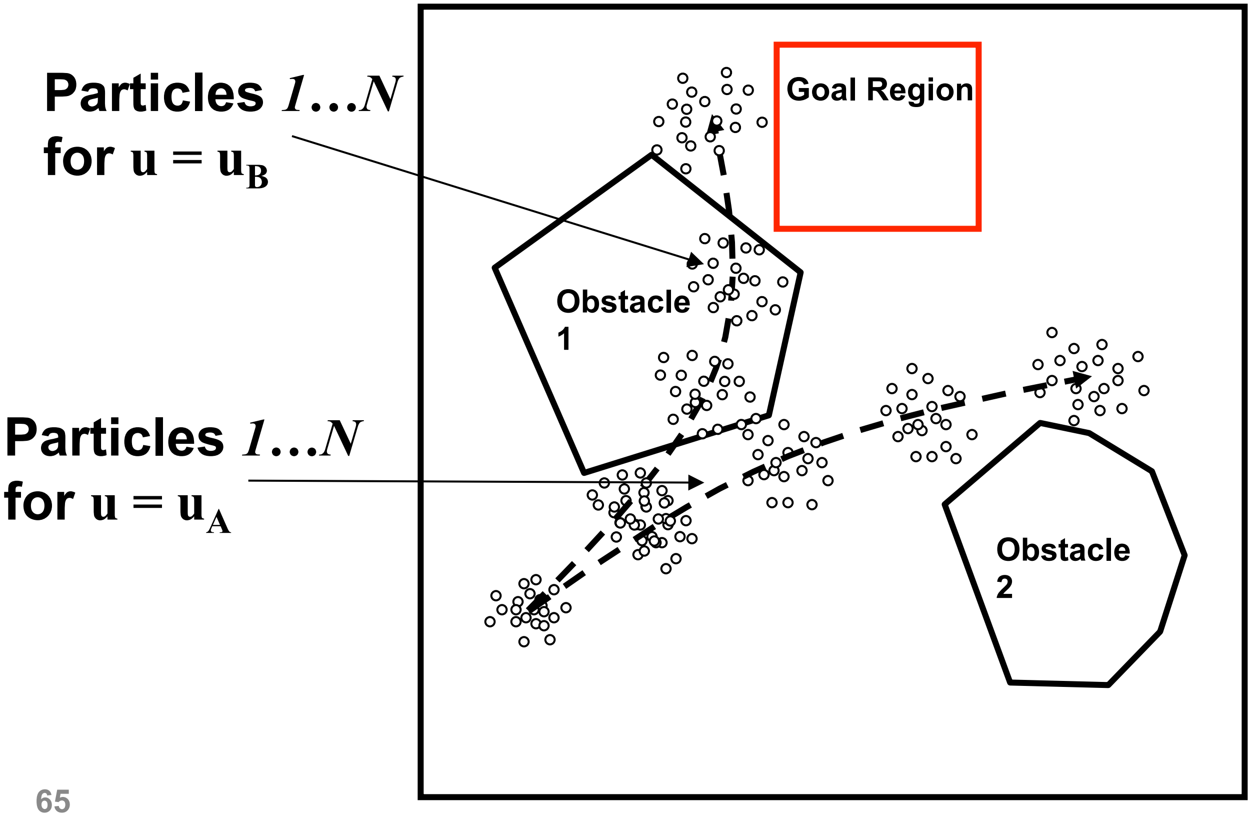


Particle Control

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

$$\mathbf{x}_t^{(i)} = f_t(\mathbf{u}_{0:t-1}, \mathbf{x}_0^{(i)}, \nu_{0:t-1}^{(i)})$$

$$\mathbf{x}_{0:T}^{(i)} = \begin{bmatrix} \mathbf{x}_0^{(i)} \\ \vdots \\ \mathbf{x}_F^{(i)} \end{bmatrix}$$

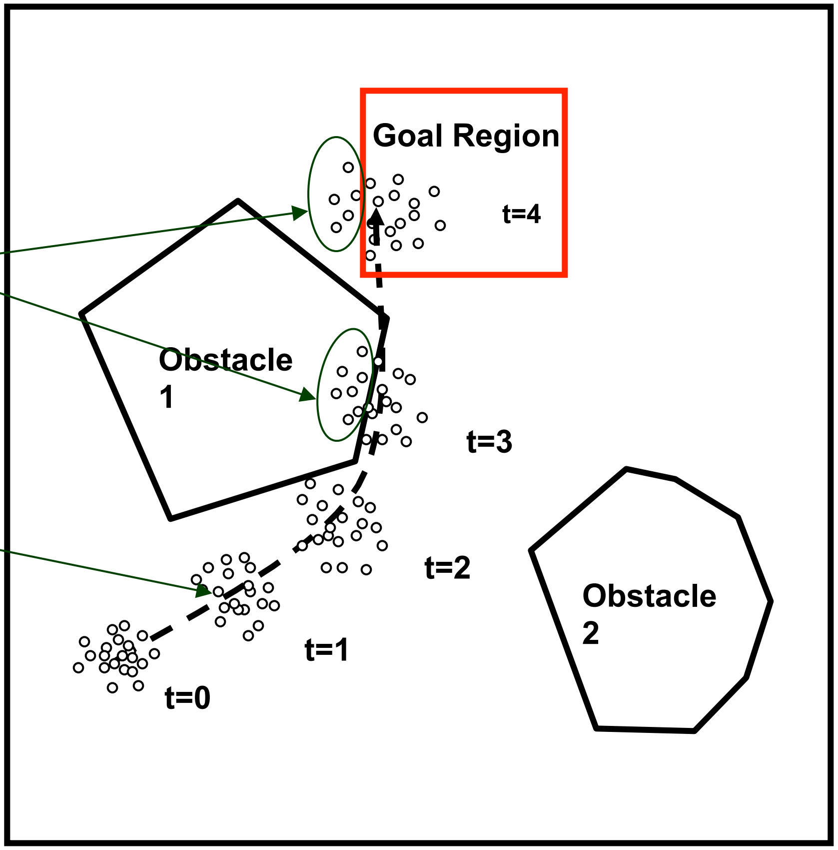


Particles *1…N* for $\mathbf{u} = \mathbf{u}_B$

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

Particle Control

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



Probability of failure approximated by the fraction of failing particles.

.

Sample mean approximates state mean.

True expectation approximated by sample mean of cost function:

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

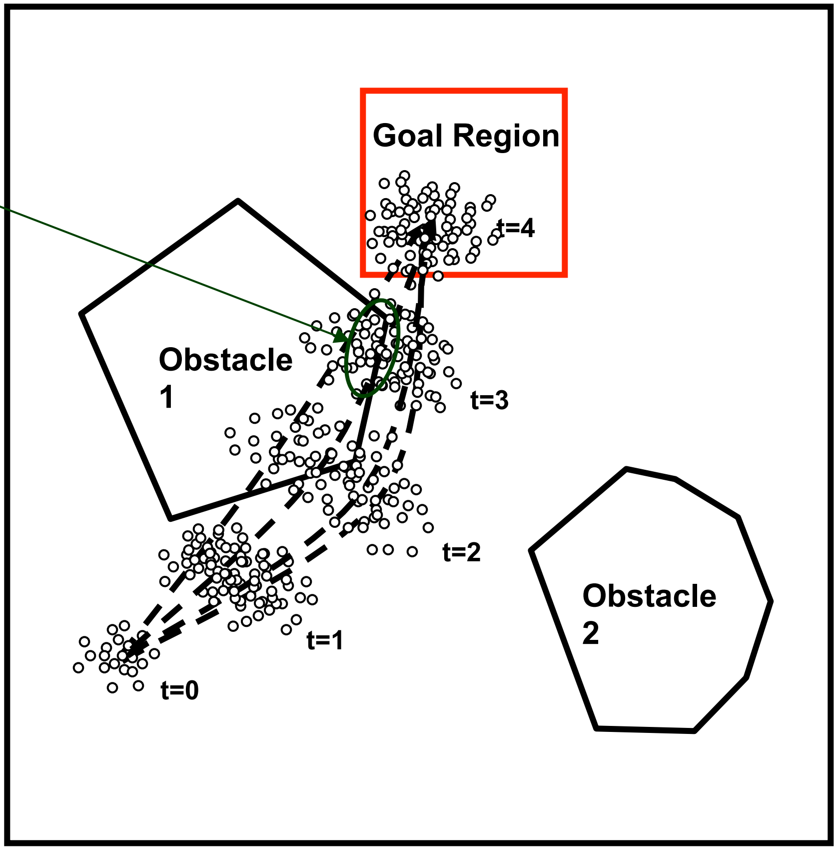
$$\approx \frac{1}{N}\sum_{i=1}^{N} h(\mathbf{u}_{0:F-1},\mathbf{x}_{1:F}^{(i)})$$

Particle Control

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

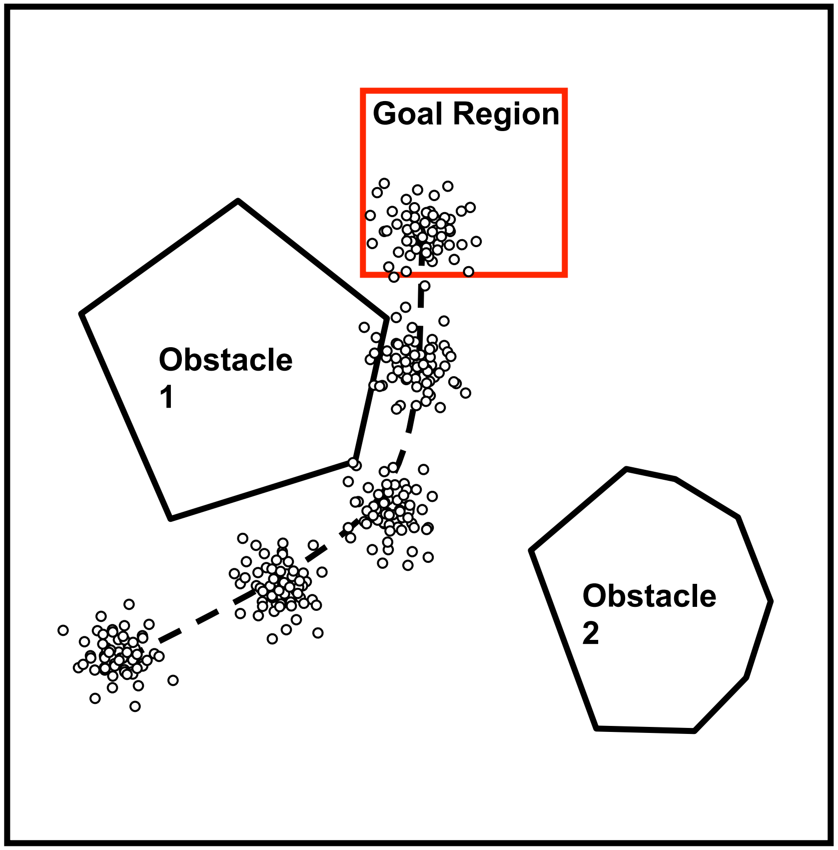
$\delta = 0.1$

10% of particles fail in optimal solution.



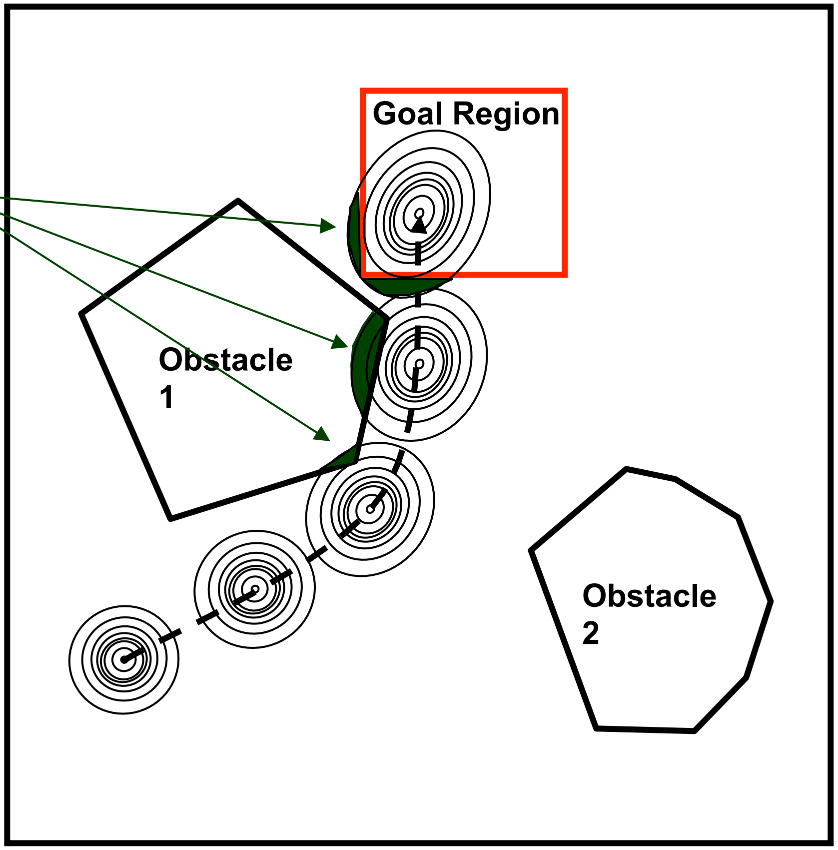
Convergence

- As N→∞, approximation becomes exact.



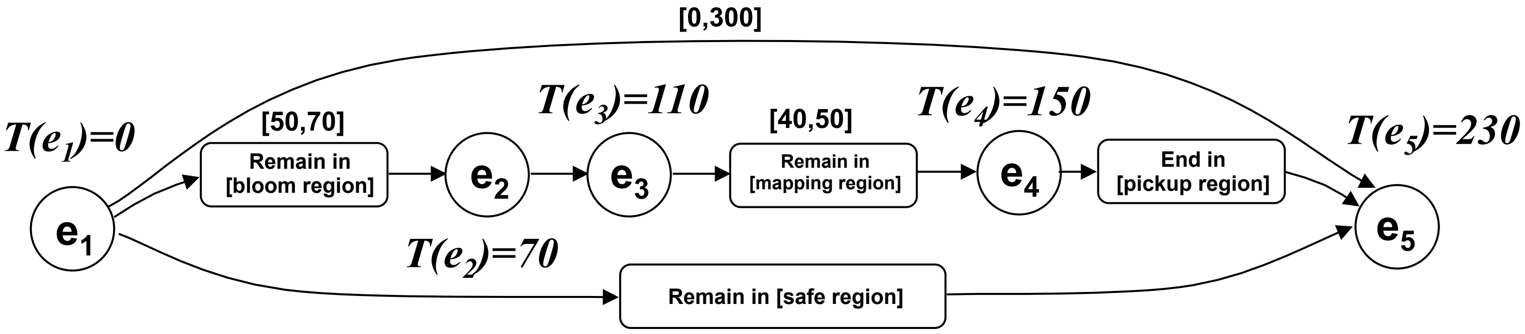
Convergence

- As N→∞, approximation becomes exact.



10% probability of failure.

MBARI AUV Science Mission



$T(e_1)=0$

e_1

[0,300]

[50,70]

Remain in [bloom region]

e_2

$T(e_2)=70$

e_3

$T(e_3)=110$

[40,50]

Remain in [mapping region]

e_4

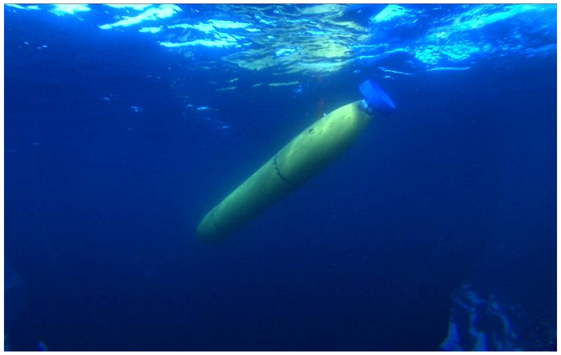
$T(e_4)=150$

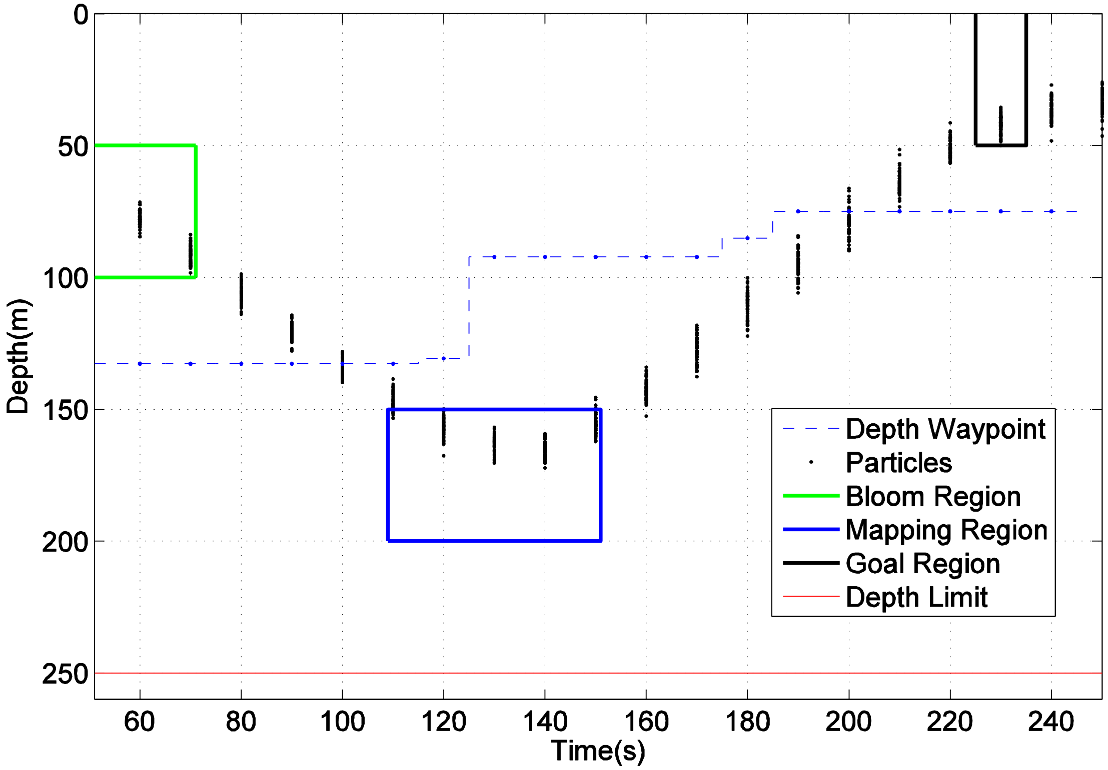
End in [pickup region]

e_5

$T(e_5)=230$

Remain in [safe region]





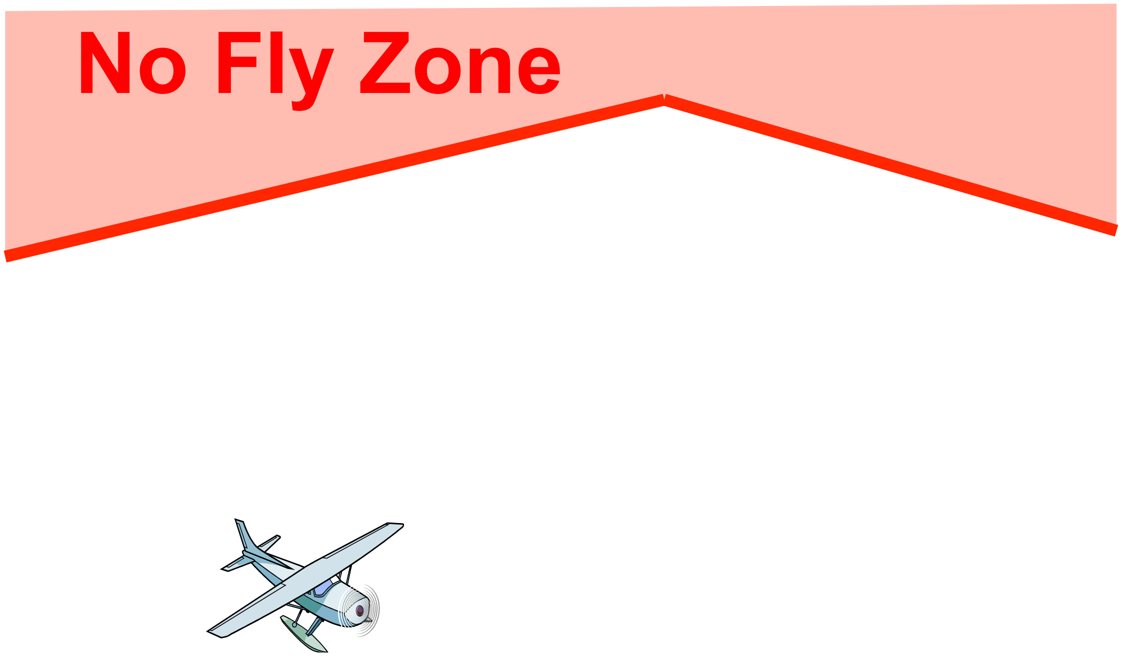
Solution Methods for Chance-Constrained Problems

- 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:

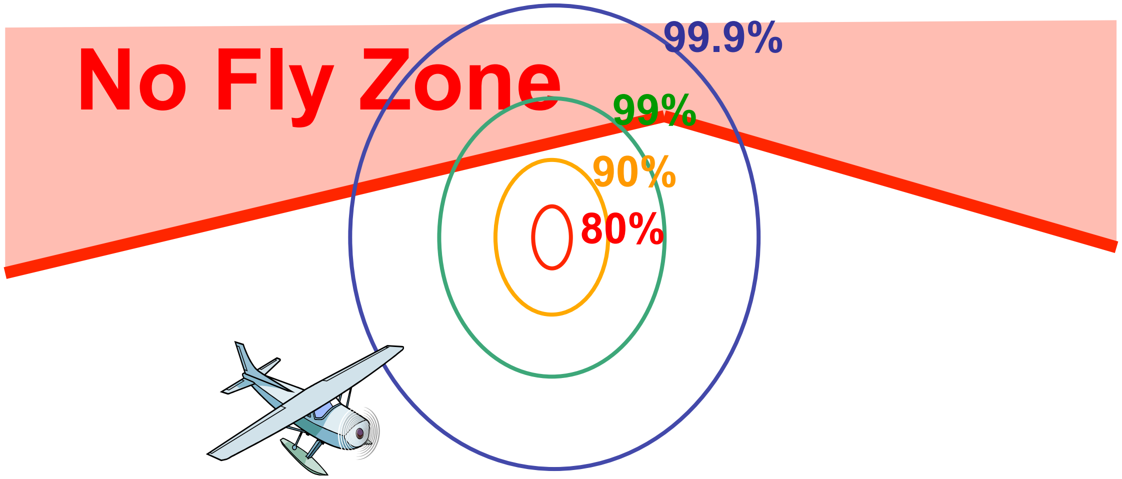
Risk < 1%



Elliptic Approximation

Chance constraint:

Risk < 1%



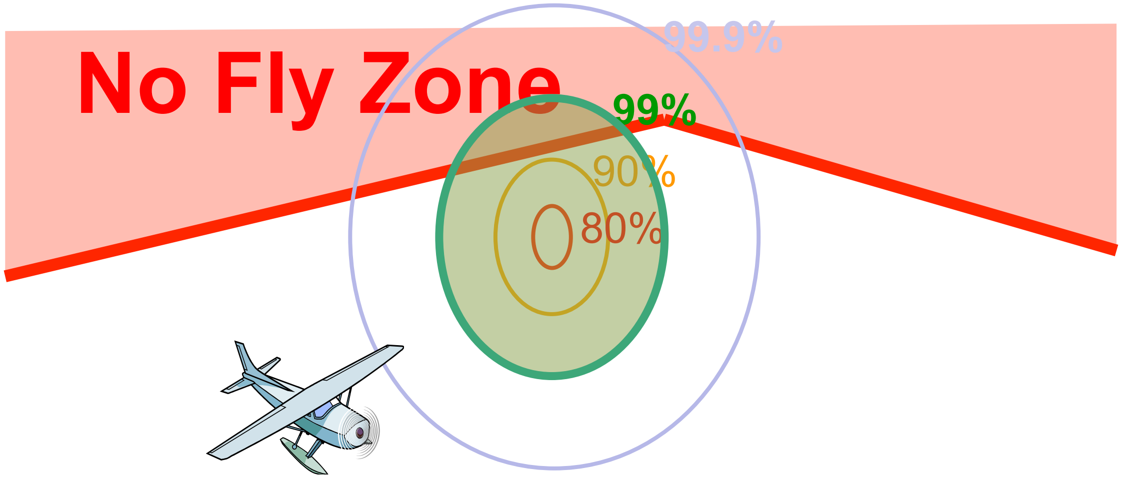
1. Specify the probability distribution of the future states as a function of control inputs.

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%

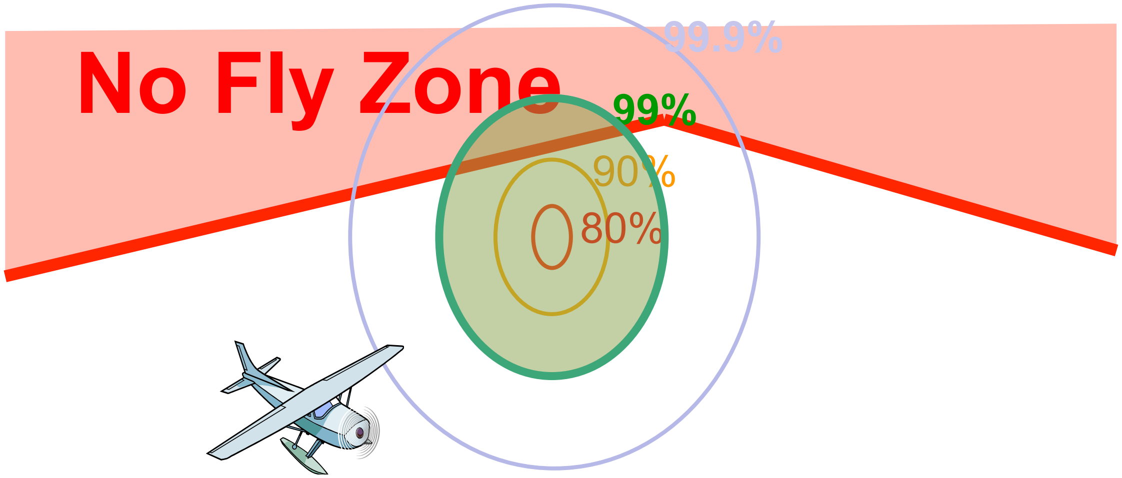


1. Specify the probability distribution of the future states as a function of control inputs.
2. Find a 99% probability ellipse.

Elliptic Approximation

Chance constraint:

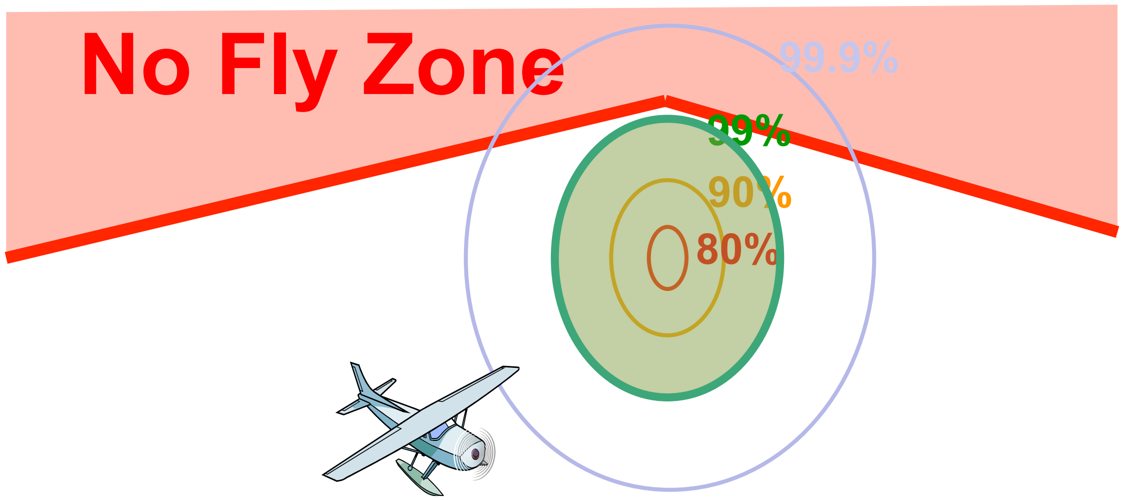
Risk < 1%



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

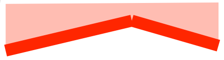
Conservatism of Elliptic Approximation

Issue: often *very* conservative



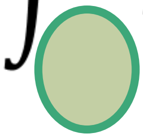
Real probability of failure

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



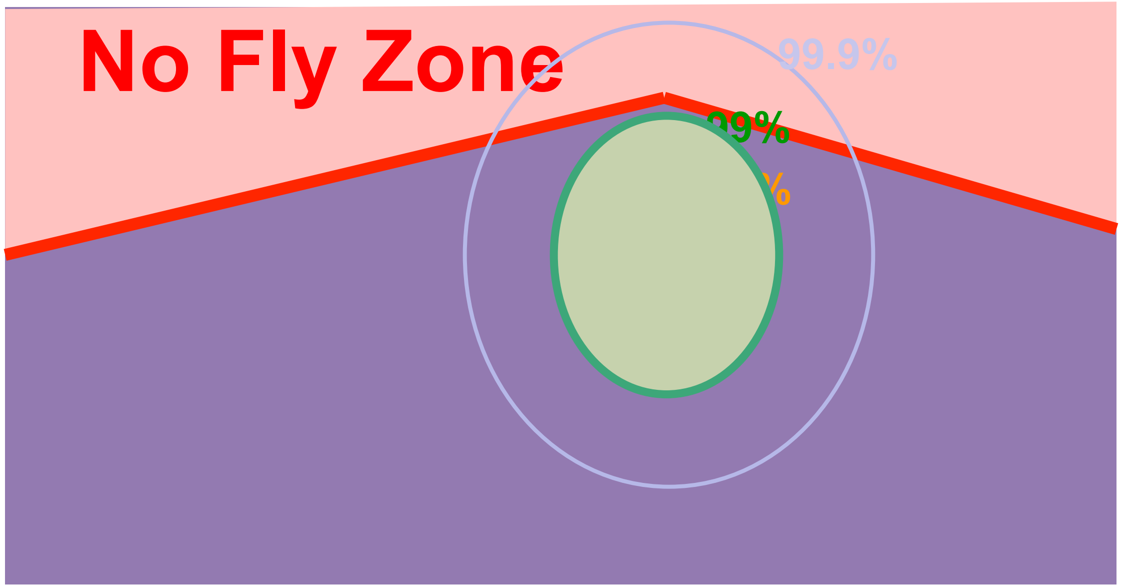
Probability density function

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

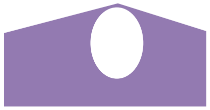


Conservatism of Elliptic Approximation

Issue: often *very* conservative.



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

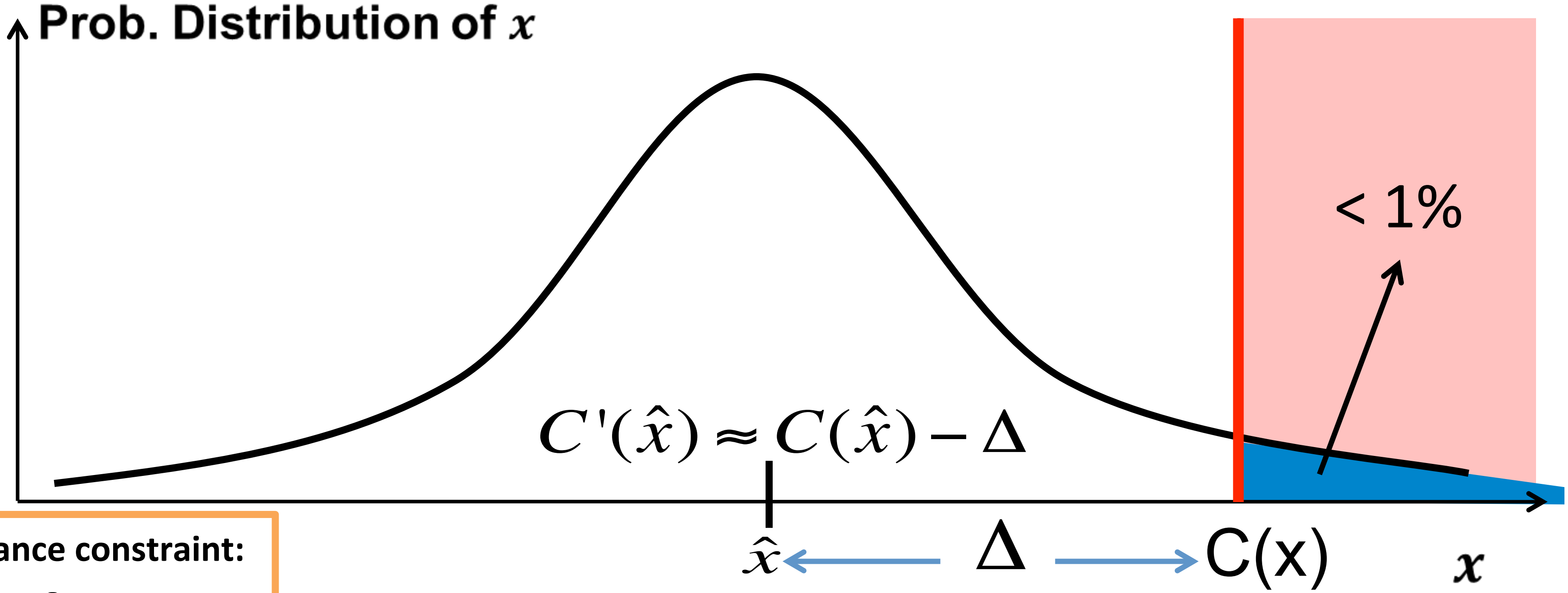


Solution Methods for Chance-Constrained Problems

- 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**

Risk-Allocation Approach: Overview

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’.



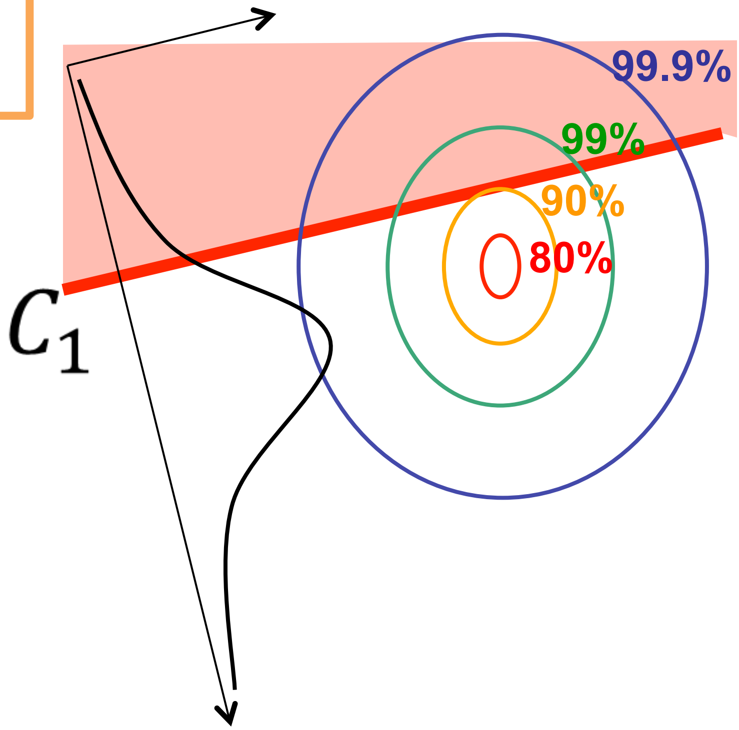
Chance constraint:

Risk < 1%

Risk-Allocation Approach: Overview

Chance constraint:

Risk < 1%



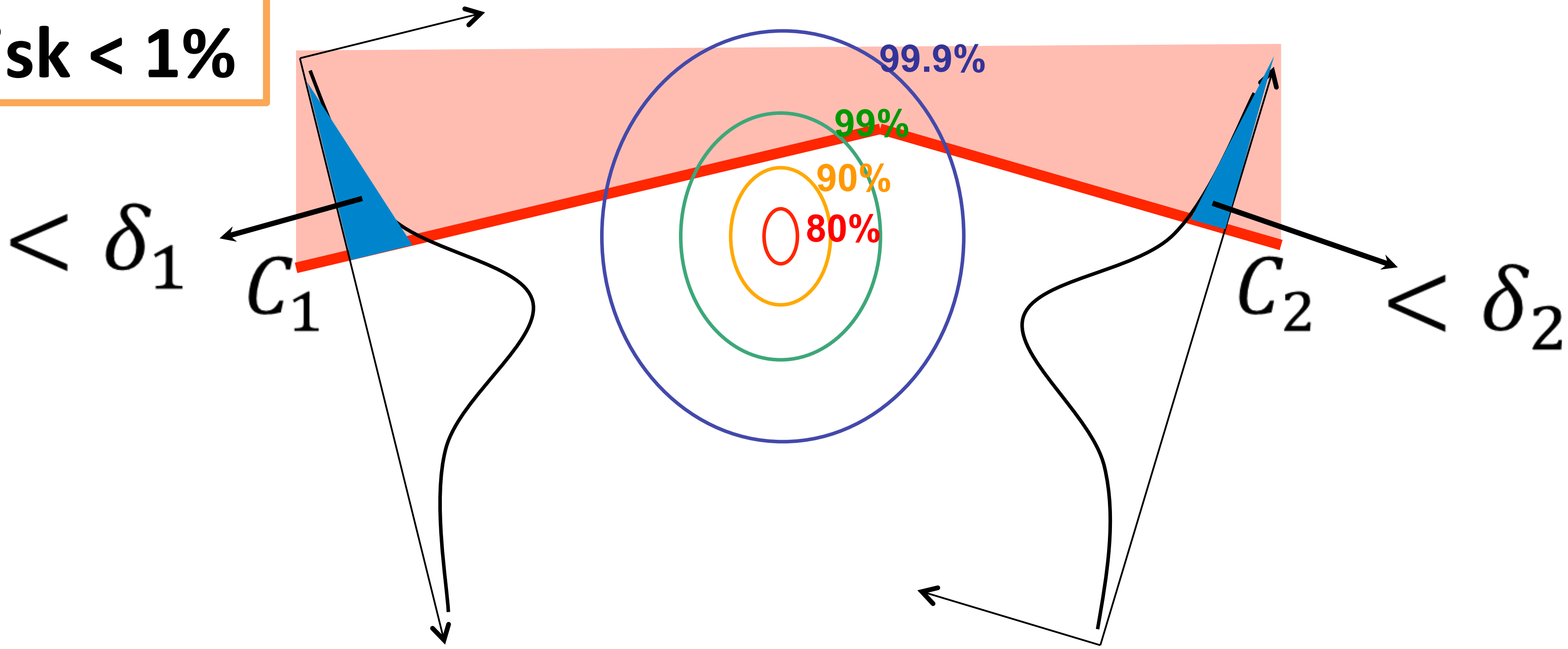
Idea 2: Generalize to a single constraint over an N-dimensional random variable, by projecting its distribution onto the axis perpendicular to the constraint boundary.

Model-based Embedded & Robotic Systems

Risk-Allocation Approach: Overview

Chance constraint:

Risk < 1%

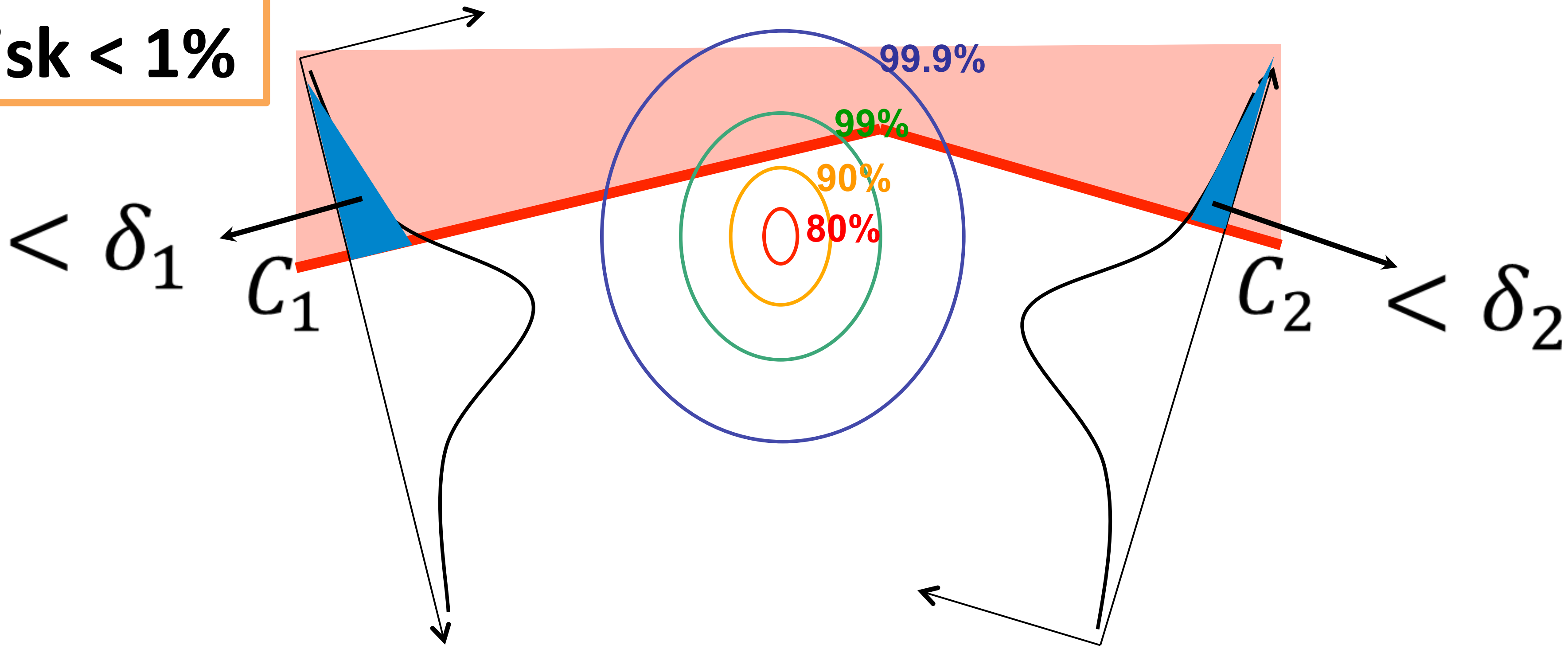


Idea 3: Generalize to a joint chance-constraint over multiple constraints C_1, C2, by distributing risk.

Risk-Allocation Approach: Overview

Chance constraint:

Risk < 1%



Find a solution such that:

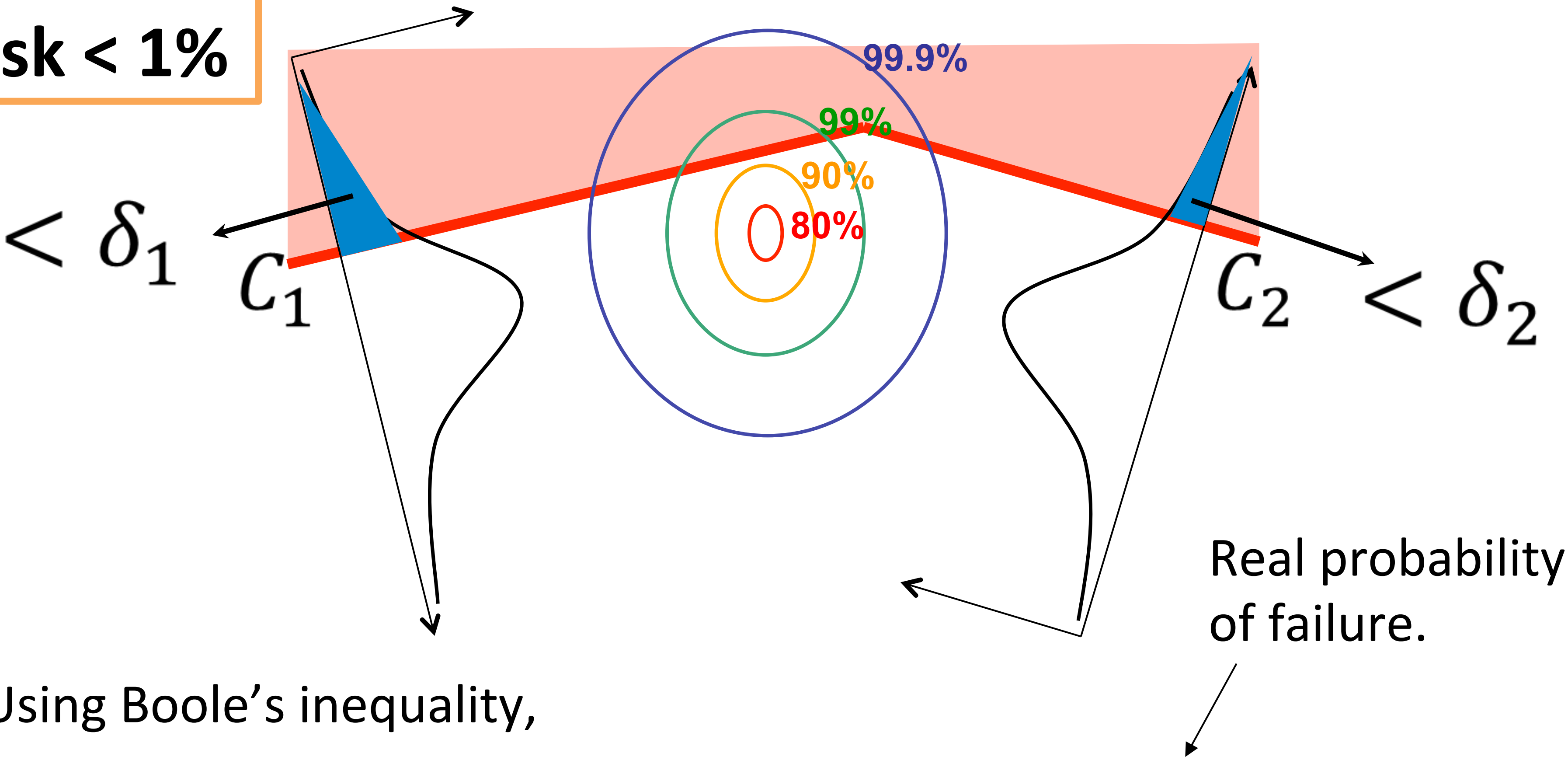
1. Each constraint C_i takes less than δ_i risk.
2. $\Sigma_i \delta_i \leq 1\%$

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

Risk-Allocation Approach: Overview

Chance constraint:

Risk < 1%



Real probability of failure.

Using Boole's inequality,

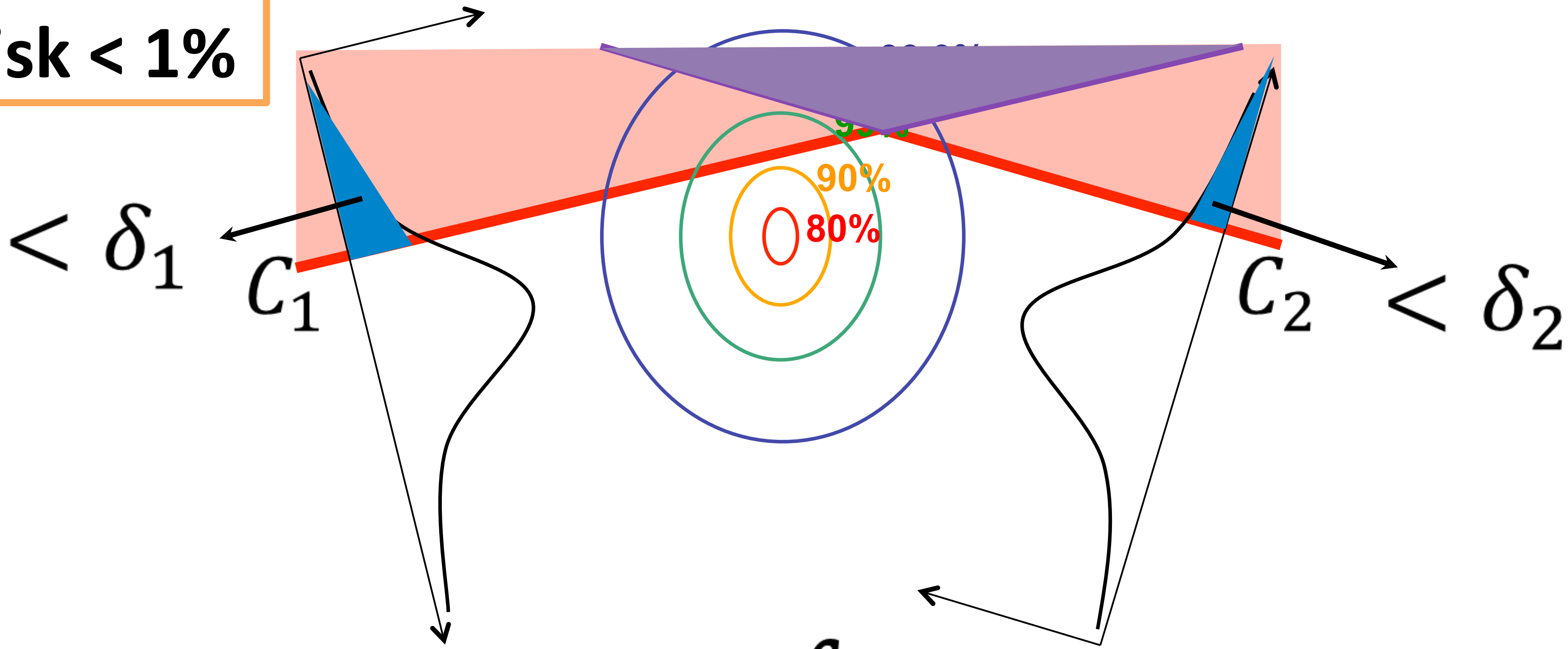
$$1\% \geq \Pr[F_1] + \Pr[F_2] \geq \Pr[F_1 \cup F_2]$$

where F_i is an event in which C_i is violated.

Risk-Allocation Approach: Conservatism

Chance constraint:

Risk < 1%



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

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

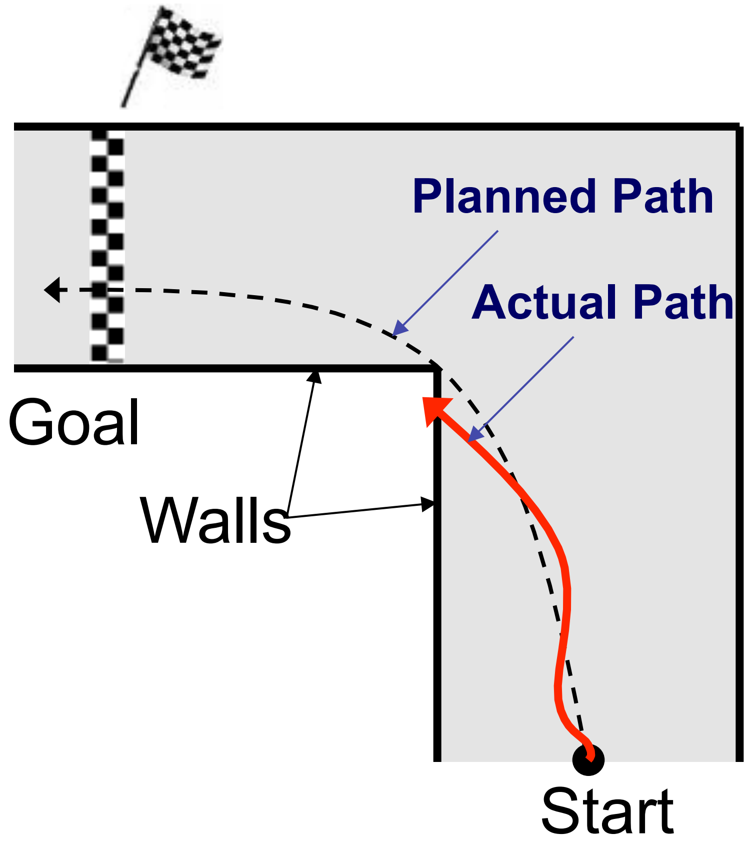
Managing Vehicles using Risk Allocation

Ono & Williams, JAIR13
Yu & Williams, IJCAI13

Outline

- Goal-directed, Model-Predictive Control
- Stochastic Optimization
- Iterative Risk Allocation
- Optimal Risk Allocation
- Appendix: Multi-agent Risk Allocation

Example: Race Car Path Planning

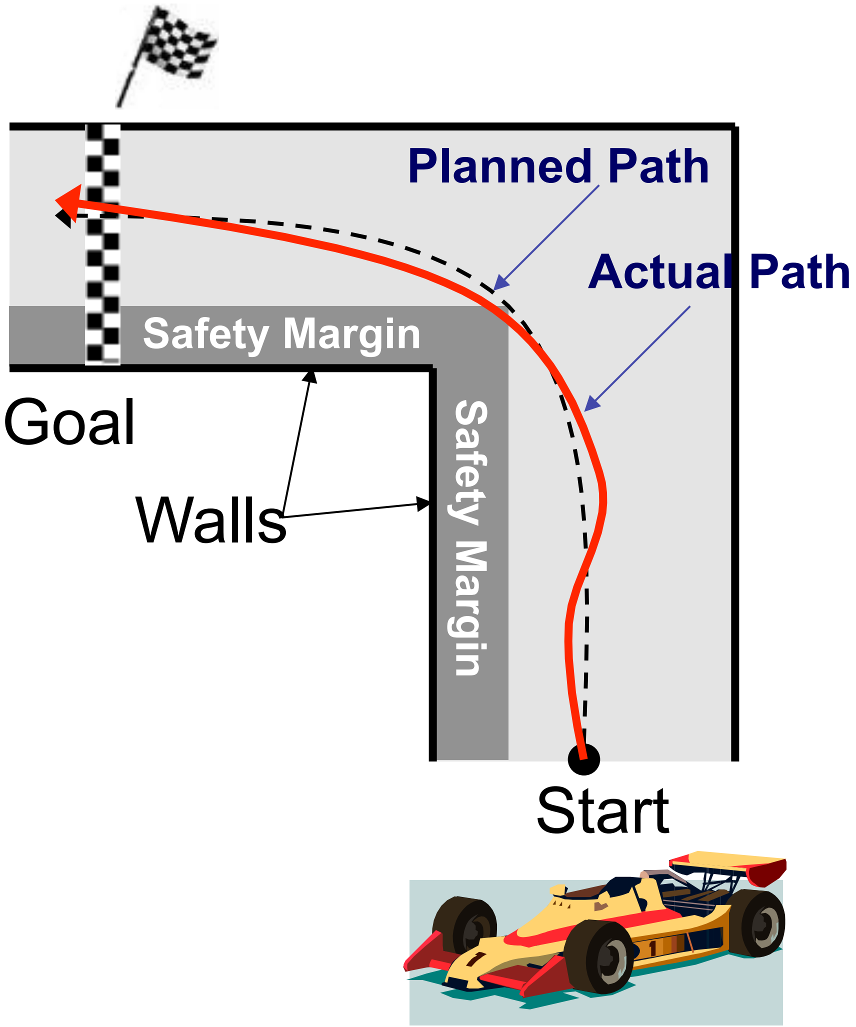


- A race car driver wants to go from the start to the goal as fast as possible.
- Actual path may differ from the planned path due to uncertainty.
- Crashing into the wall may kill the driver.
- Driver wants a probabilistic guarantee:

$$P(\text{crash}) < 0.1\%$$

  - **Chance constraint.**

Idea: Plan Nominal Path using Safety Margin



Problem

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

Risk bound

Approach:

1. Set **safety margin** that guarantees that the risk bound is satisfied.
2. Plan optimal nominal path within safety margin.

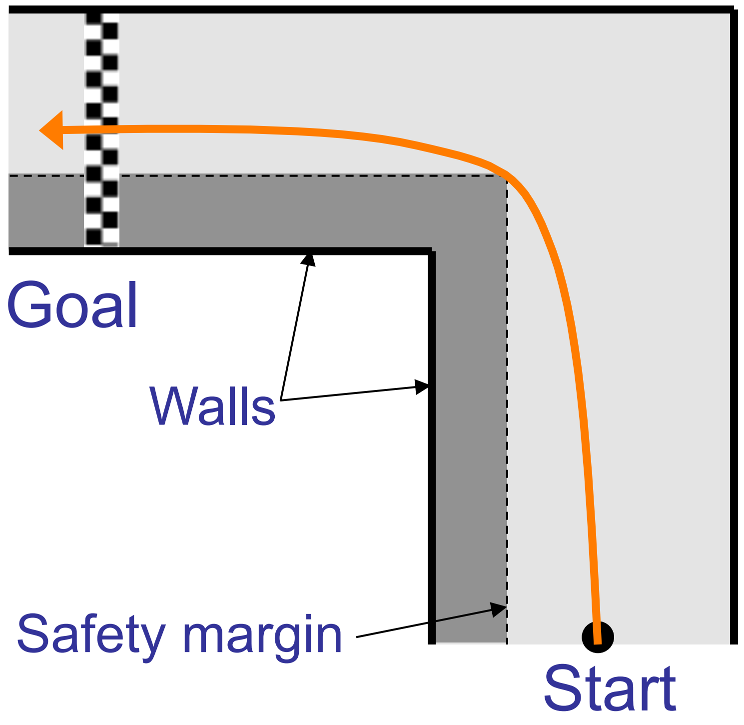
Simple Method:

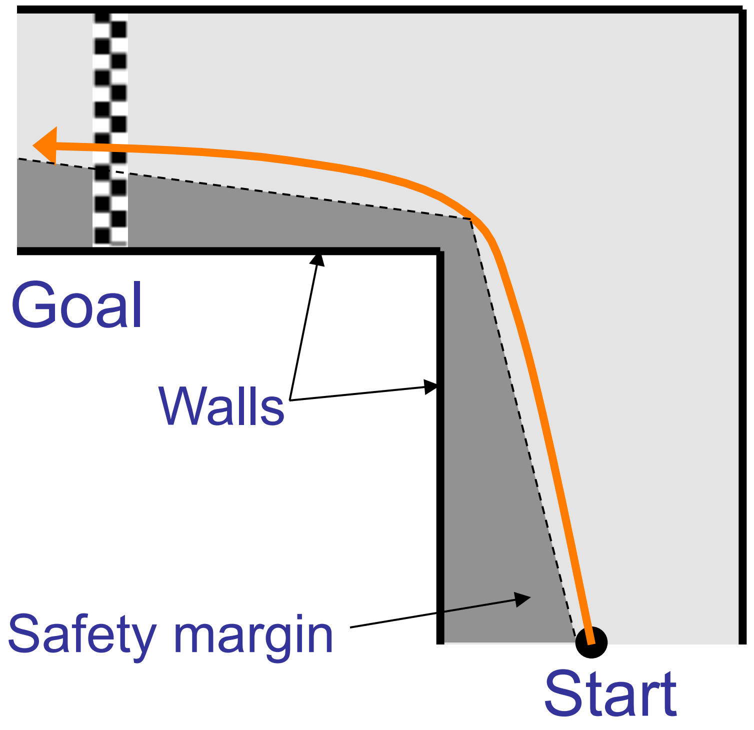
Uniform risk allocation.

Not All Safety Margins are Equal

Uniform width

Non-uniform width





Longer path

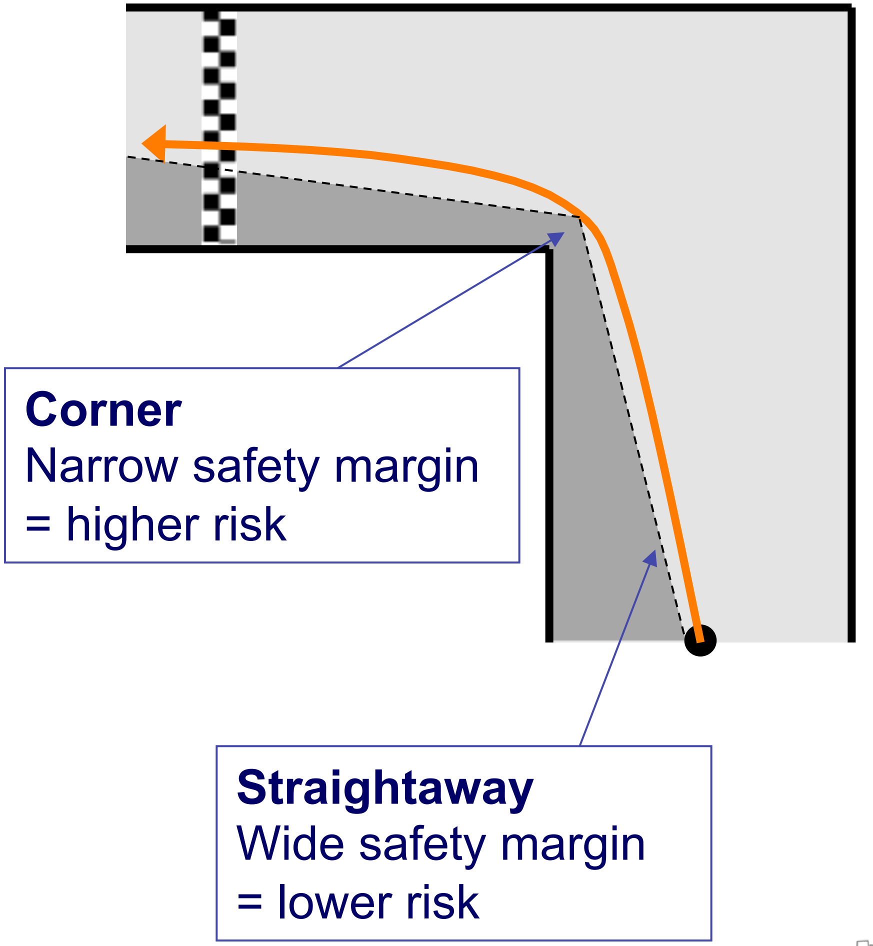
Shorter path

Idea: Design the Optimal Safety Margin by Allocating Risk

- Added risk at the corner shortens the path more than the same amount of risk at the straightaway.
  - **Sensitivity** of path length to changes in risk **is higher near the corner.**

Risk Allocation:

- Find an allocation of risk to constraints that results in the best feasible solution.



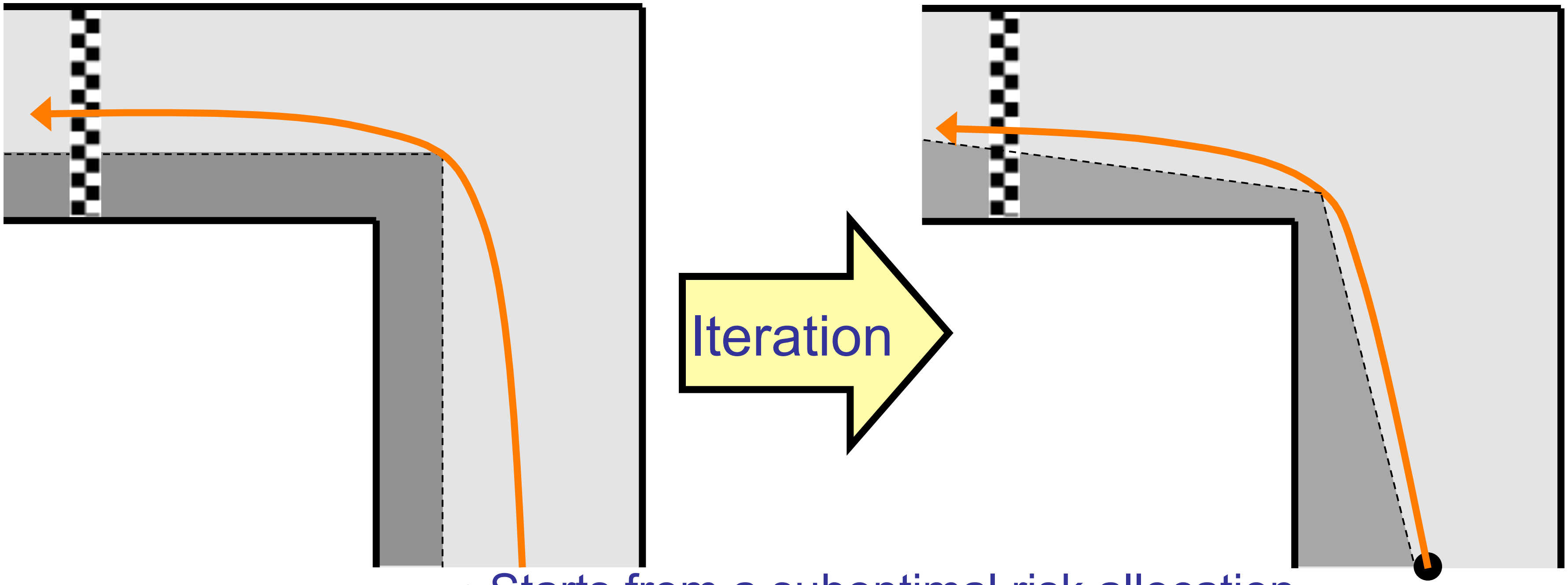
Corner
Narrow safety margin
= higher risk

Straightaway
Wide safety margin
= lower risk

Iterative Risk Allocation (IRA) Algorithm

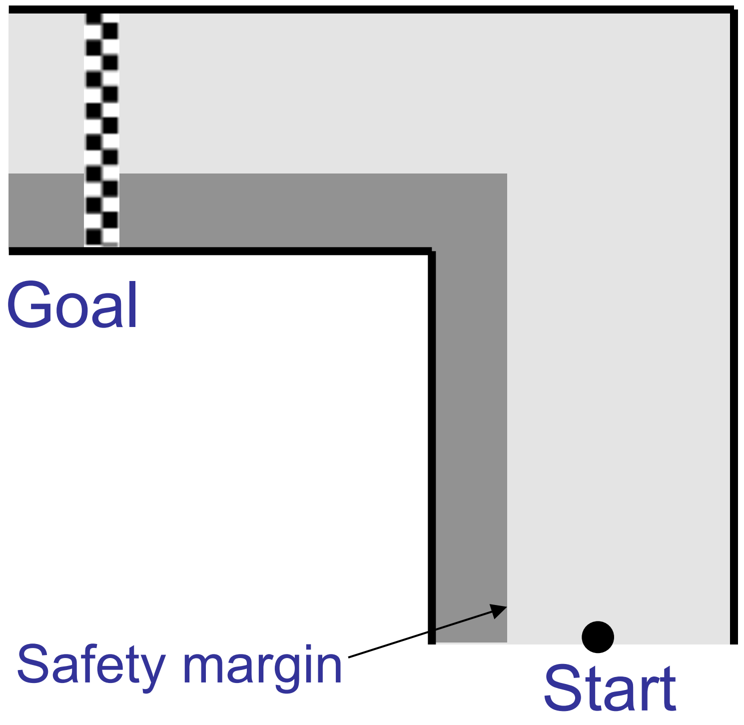
- Descent algorithm

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



- Starts from a suboptimal risk allocation.
- Improves allocation at each iteration.
- But does not guarantee convergence.

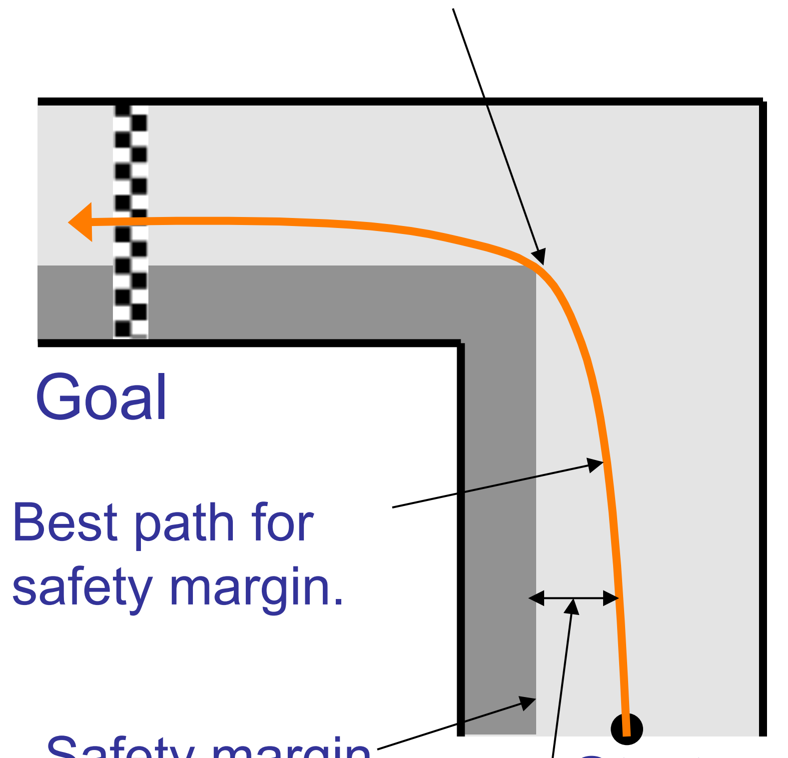
Iterative Risk Allocation Algorithm



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



No gap = Constraint is ***active.***

Goal

Best path for safety margin.

Safety margin

Start

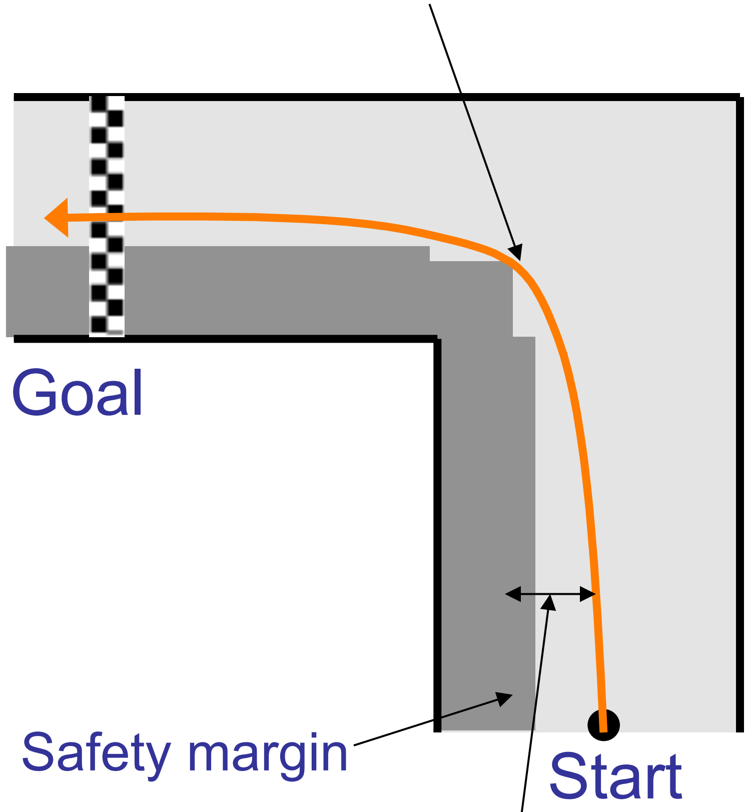
Gap = constraint is ***inactive.***

Algorithm IRA

```
Initialize with arbitrary risk allocation.
Loop
    Compute the best available path given the current risk allocation.
    Decrease the risk where the constraint is inactive.
    Increase the risk where the constraint is active.
End loop
```

Iterative Risk Allocation Algorithm

No gap = Constraint is ***active.***

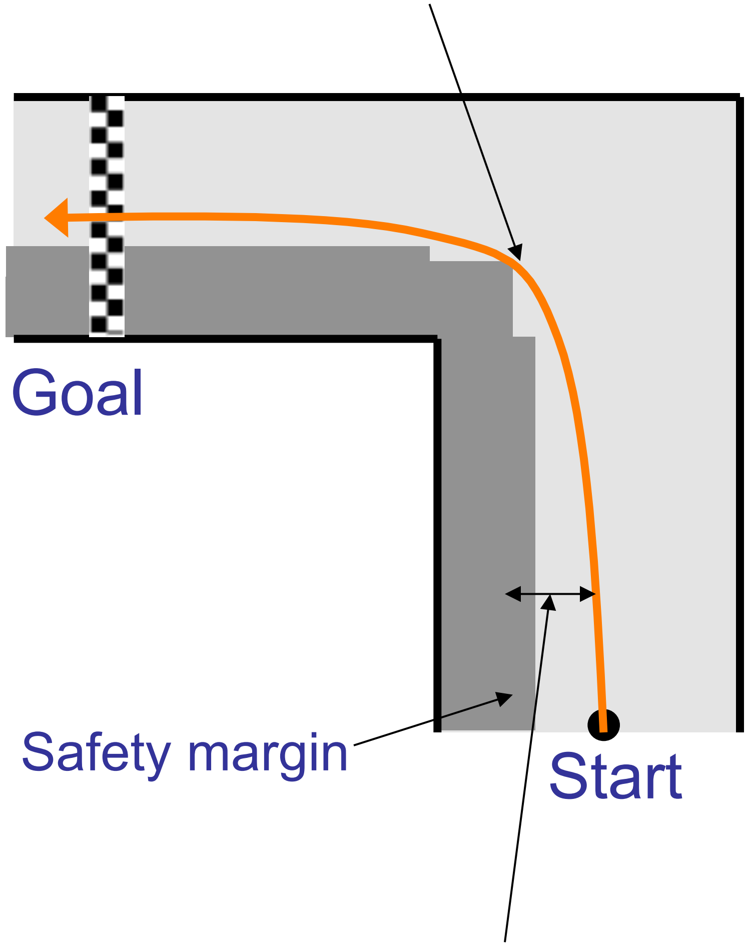


Gap = constraint is ***inactive.***

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



No gap = Constraint is ***active.***

Goal

Safety margin

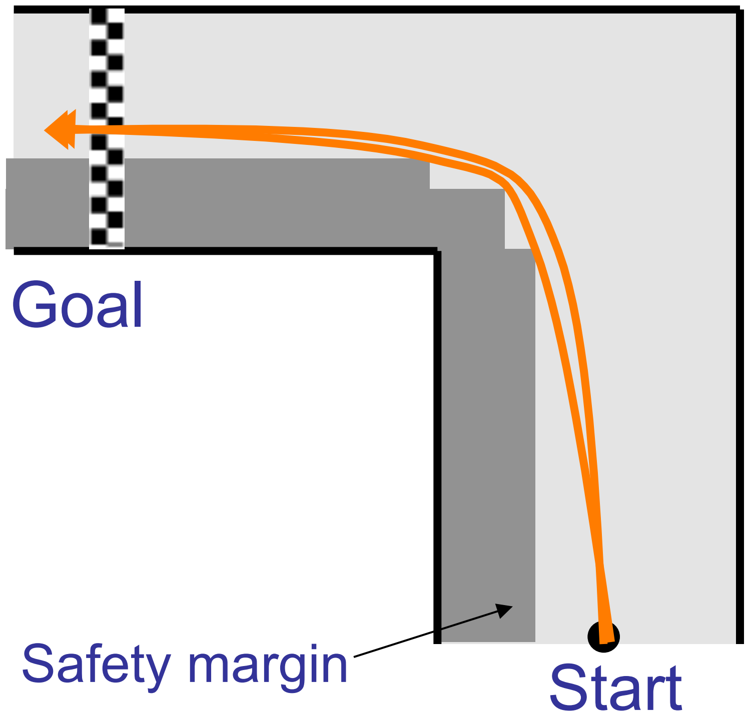
Start

Gap = constraint is ***inactive.***

Algorithm IRA

```
Initialize with arbitrary risk allocation.
Loop
    Compute the best available path given the current risk allocation.
    Decrease the risk where the constraint is inactive.
    Increase the risk where the constraint is active.
End loop
```

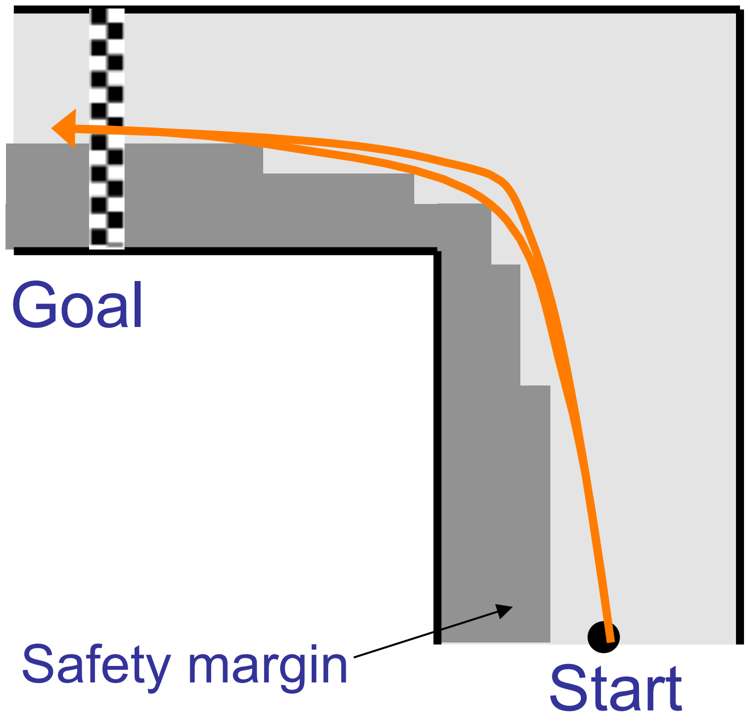
Iterative Risk Allocation Algorithm



Algorithm IRA

```
Initialize with arbitrary risk allocation.
Loop
    Compute the best available path given the current risk allocation.
    Decrease the risk where the constraint is inactive.
    Increase the risk where the constraint is active.
End loop
```

Iterative Risk Allocation Algorithm



What Remains:

- Mathematical formulation:
  - Reformulating stochastic to deterministic constraints.

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
```

Comparison

- Approaches
  - ***Elliptic Approximation:*** uses very conservative approximation of joint chance constraint.
  - ***Sampling:*** approximates probability distribution by samples.
- Risk allocation results in near-optimal solution with significantly less computation time than sampling.

$$\Delta = 0.1$$

	Risk allocation	Elliptical set conversion	Sampling
Resulting probability of constraint violation	**0.097**	2.0×10^{-6}	0.0022
Objective function value	**3.15**	5.26	3.76
Computation time [sec]	3.38	**1.76**	1.41×10^{4}

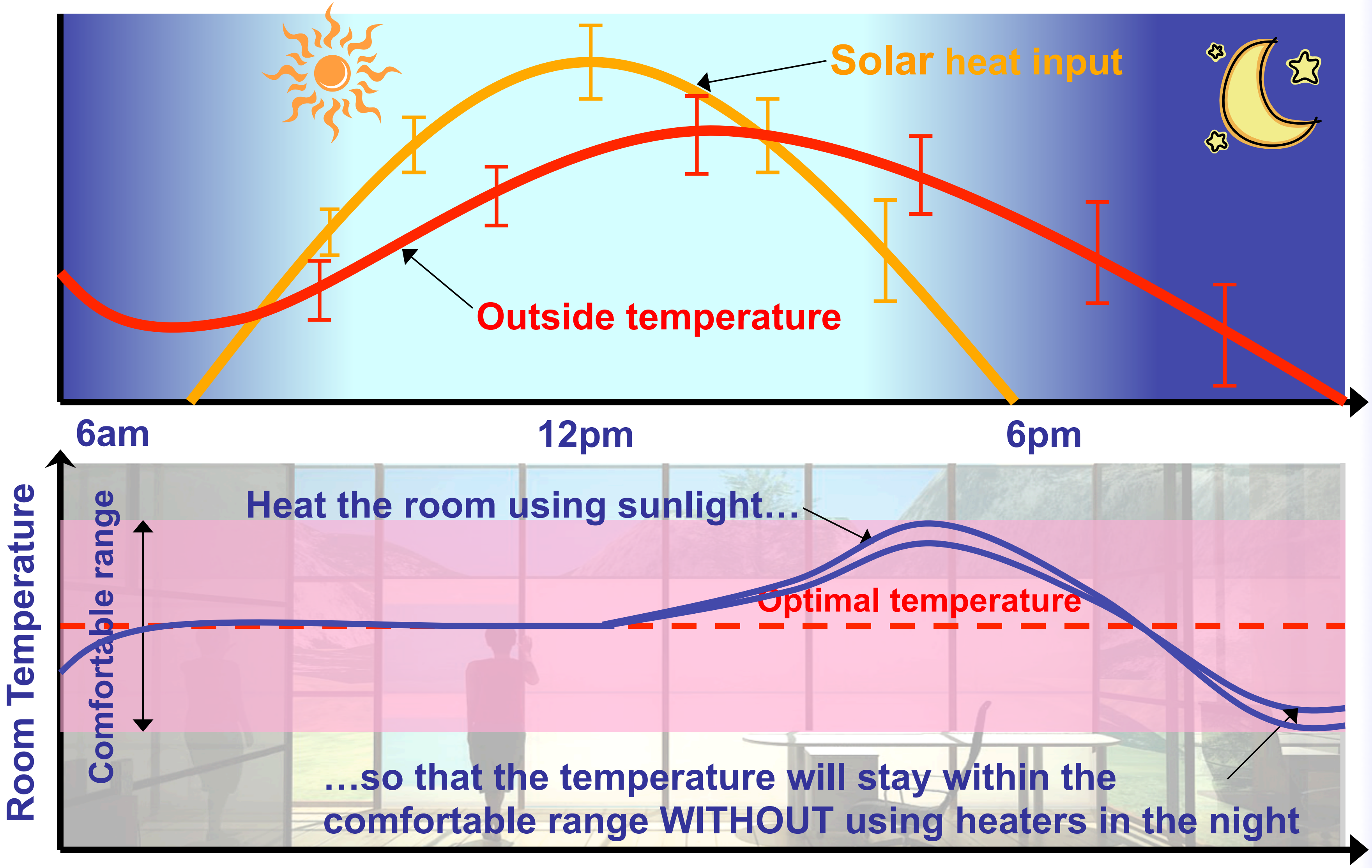
$$J=\sum_{t=1}^{T}\left\|x_t\right\|_2 \quad A=\begin{bmatrix}1 & 1\\ -0.5 & 0\end{bmatrix} \quad B=\begin{bmatrix}0\\ 0.03\end{bmatrix} \quad \Sigma_{x_0}=\begin{bmatrix}0.0001 & 0\\ 0 & 0.0001\end{bmatrix} \quad \Sigma_{w}=\begin{bmatrix}0.0001 & 0\\ 0 & 0.0001\end{bmatrix} \quad T=20,\Delta=0.1$$

Test bed: Connected Sustainable Home

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

- Goal: Optimally control HVAC, window opacity, washer/dryer, e-car.
- Objective: Minimize energy cost.
- Uncertainty: Solar input, outside temp, energy supply, occupancy.
- Risk: Resident goals not satisfied; occupant uncomfortable.

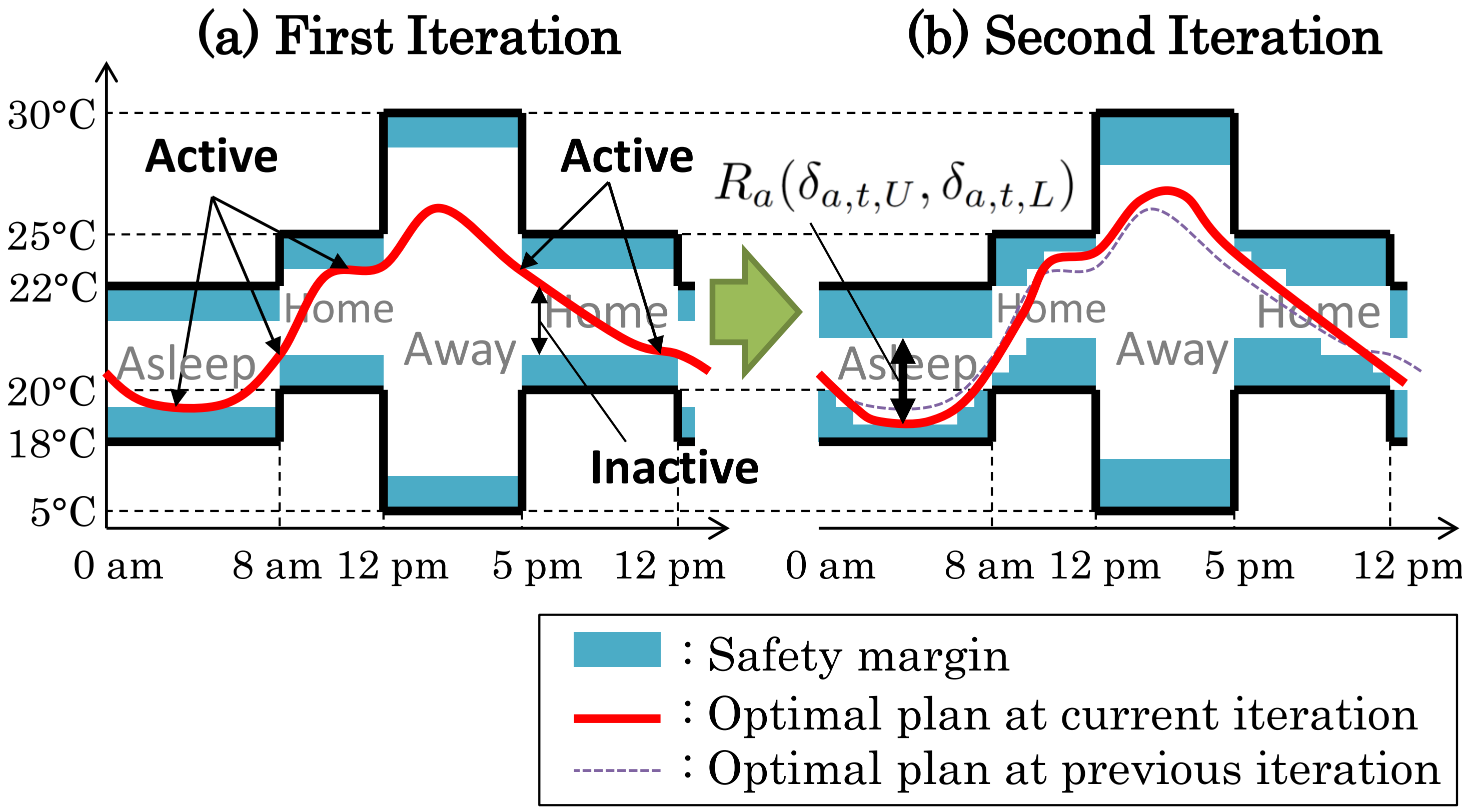
IRA-RMPC for Dynamic Window



Heat the room using sunlight…

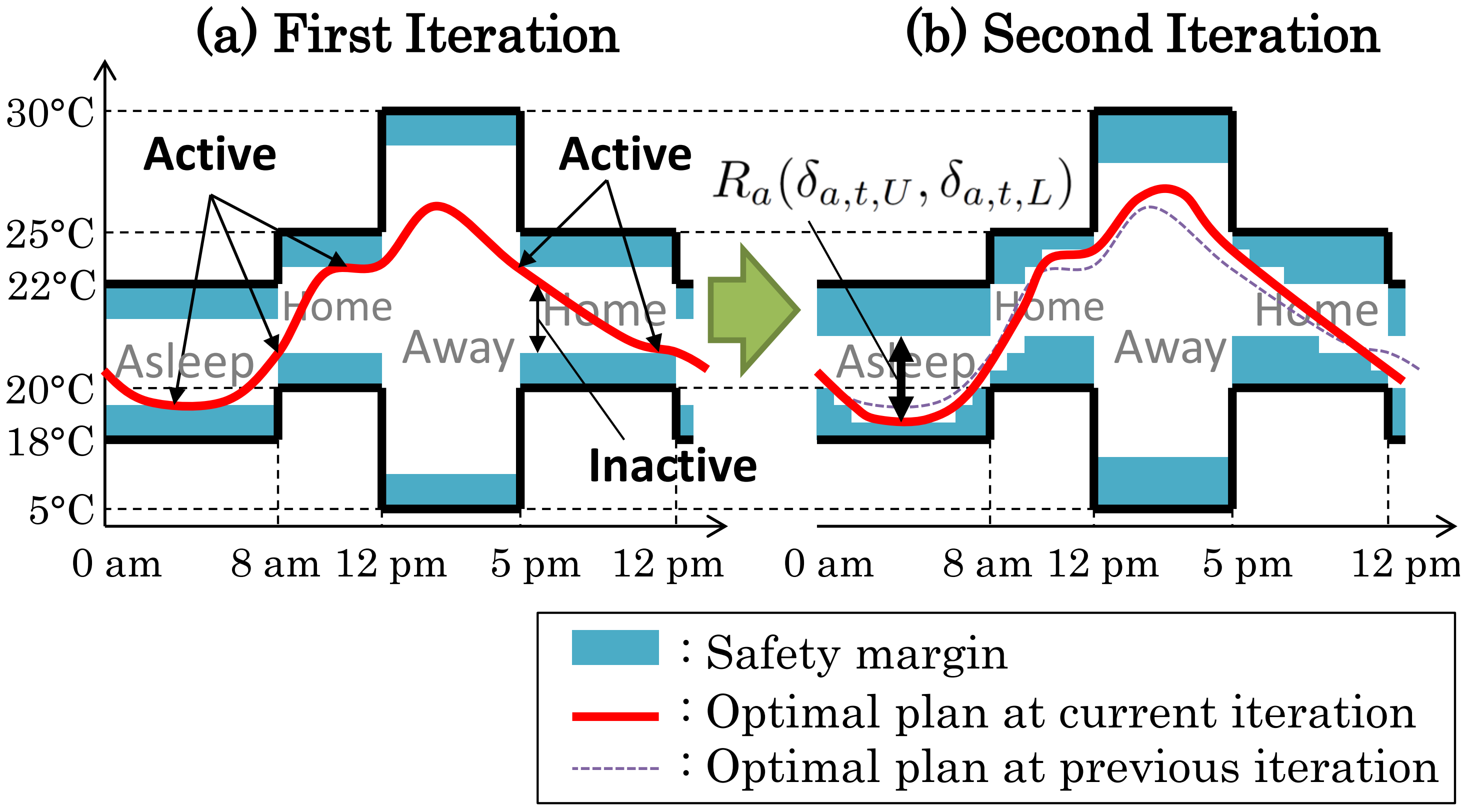
…so that the temperature will stay within the comfortable range WITHOUT using heaters in the night

Successive Risk Allocations for IRA-RMPC



Takes risk of violating resident constraints where largest energy savings are possible.

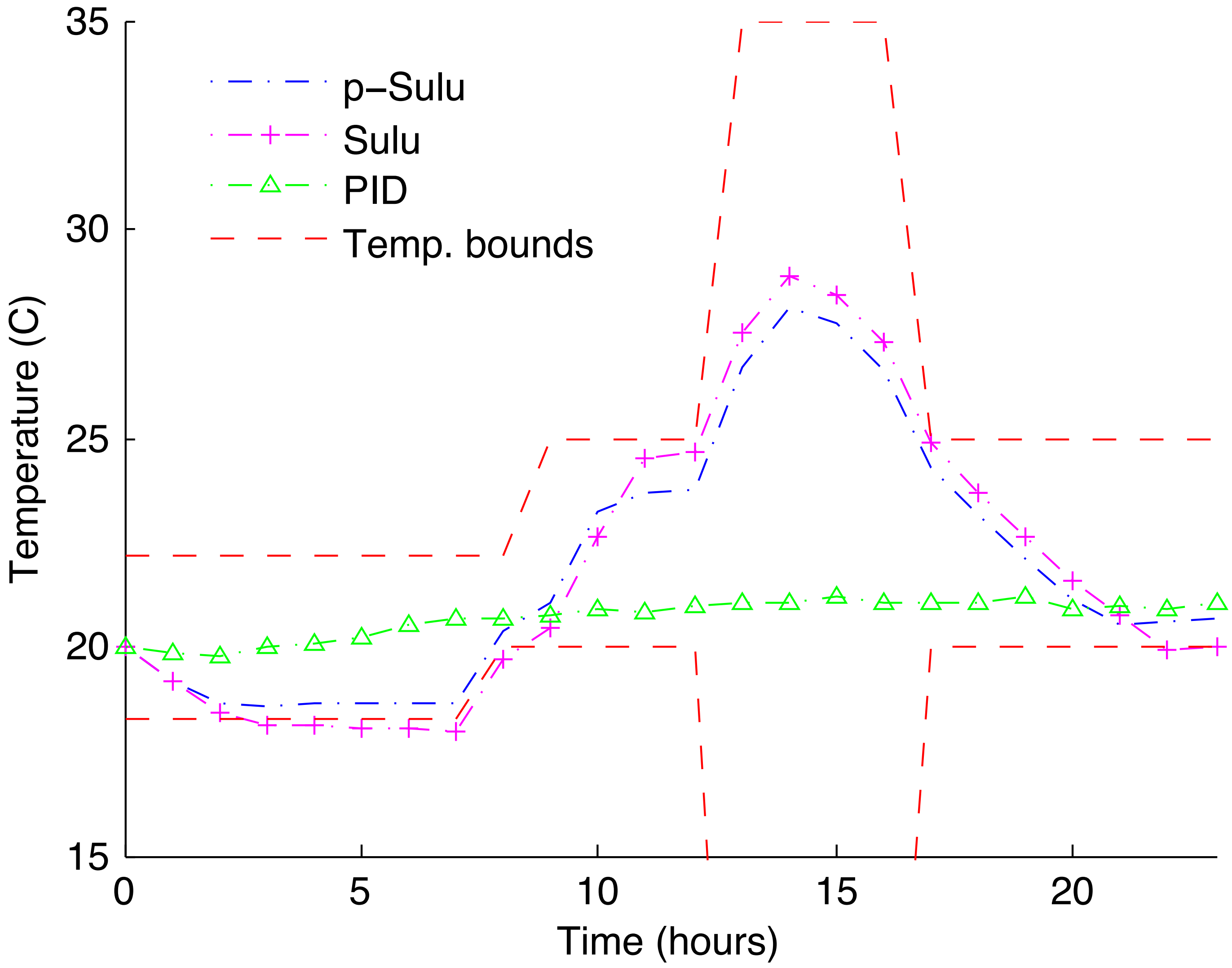
Successive Risk Allocations for IRA-RMPC



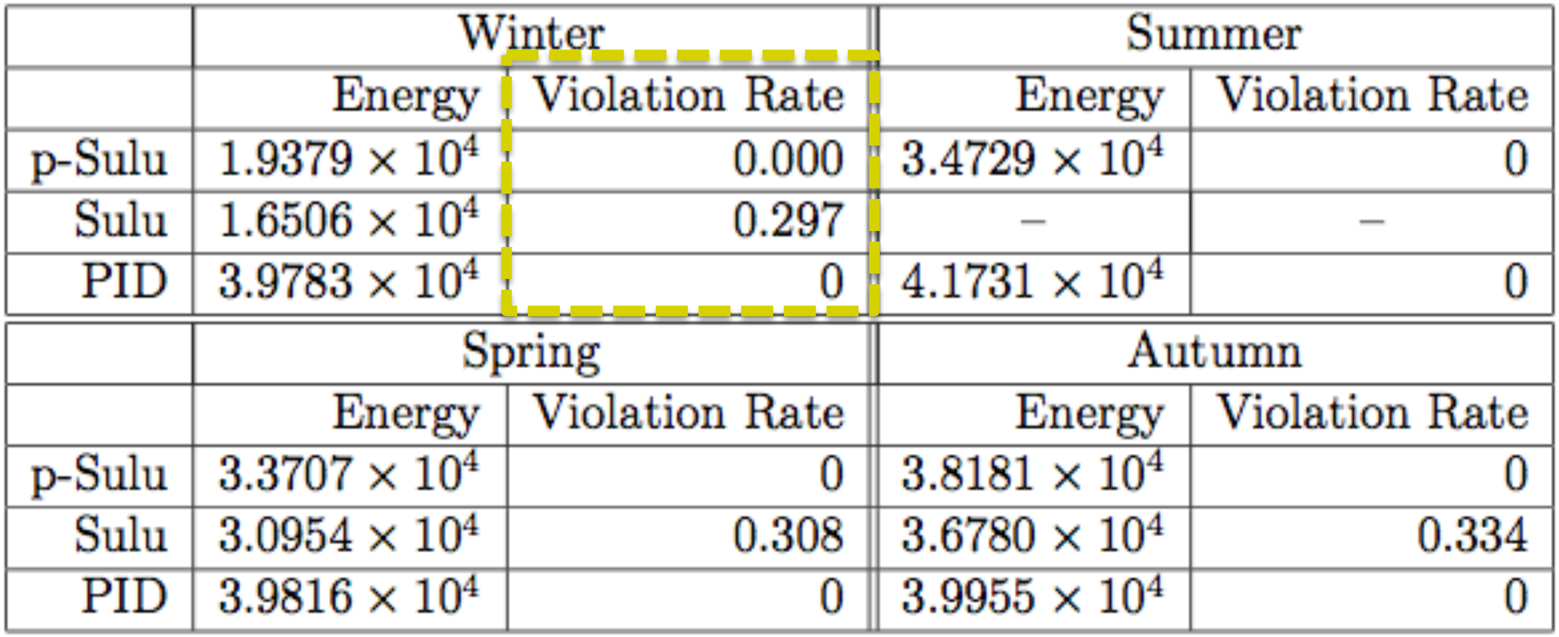
Given chance-constrained Qualitative State Plan (CC-QSP):

1. **(Re-)allocates risk.**
2. **Reformulates to deterministic QSP and calls Sulu.**
3. **Repeats.**

Results


35
30
25
20
15
0
5
10
15
20
Temperature (C)
Time (hours)
p–Sulu
Sulu
PID
Temp. bounds

Improvement in Comfort



	Winter		Summer	
	Energy	Violation Rate	Energy	Violation Rate
p-Sulu	1.9379×10^4	0.000	3.4729×10^4	0
Sulu	1.6506×10^4	0.297	–	–
PID	3.9783×10^4	0	4.1731×10^4	0

	Spring		Autumn	
	Energy	Violation Rate	Energy	Violation Rate
p-Sulu	3.3707×10^4	0	3.8181×10^4	0
Sulu	3.0954×10^4	0.308	3.6780×10^4	0.334
PID	3.9816×10^4	0	3.9955×10^4	0

- Deterministic control (Sulu): 30% comfort violations.
- Robust control (p-Sulu): near 0% violations.

Outline

- Goal-directed, Model-Predictive Control
- Stochastic Optimization
- Iterative Risk Allocation
- Optimal Risk Allocation
  - Stochastic Linear Programs
  - Disjunctive Linear Programs
  - Probabilistic Sulu
- Appendix: Multi-agent Risk Allocation

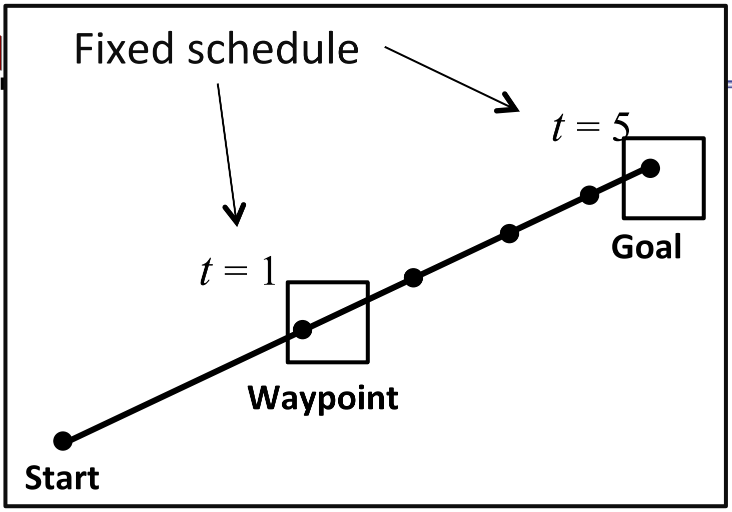
Finding Optimal Risk Allocations

Given that the Boole’s inequality approximation has been performed.

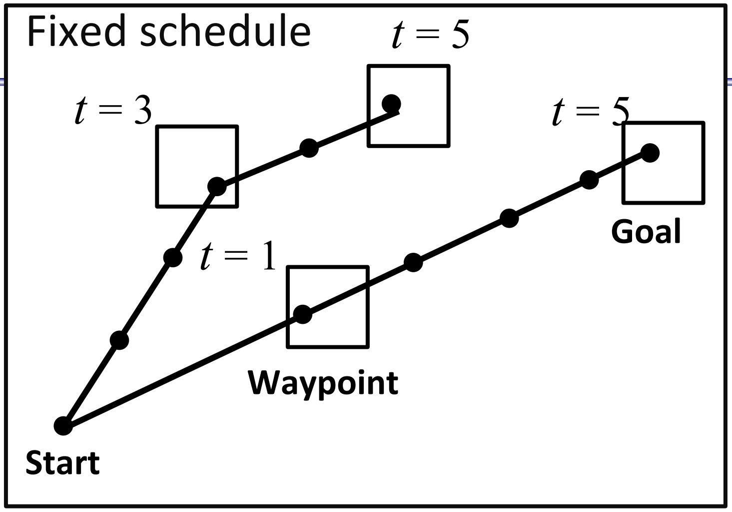
Idea:

1. Formulate optimal risk allocation as a stochastic program.
2. Map to deterministic (non-)convex program, with risk and control variables as decision variables.
3. Solve exactly using deterministic solver.

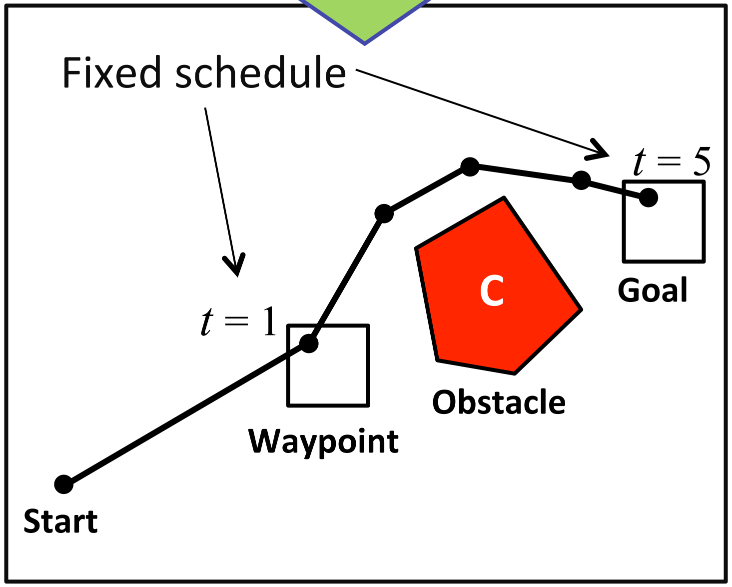
Problems



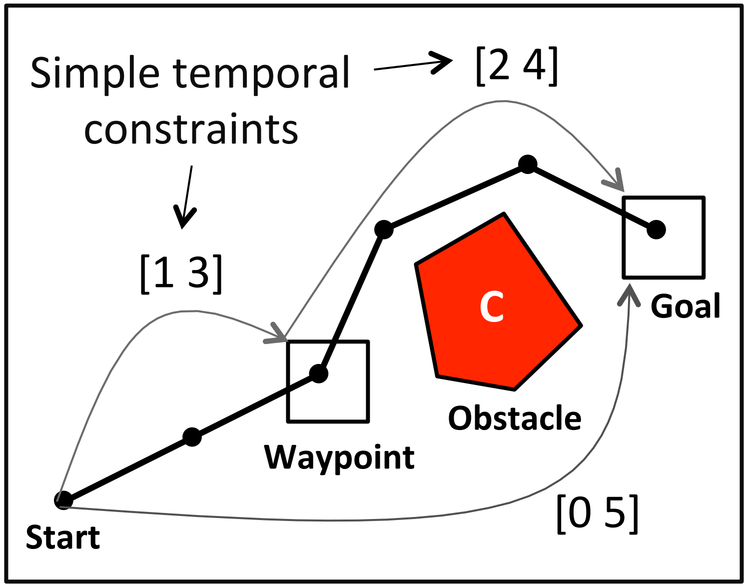
Convex, single agent



Convex, multi-agent



Non-convex, single agent



Non-convex, flexible schedule, single agent

Chance-Constrained FH Optimal Control

Model-based Embedded & Robotic Systems

Instance of a Stochastic Linear Program with Gaussian Noise

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

s.t.

Stochastic dynamics

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

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

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

Risk bound

(Upper bound of the probability of failure)

Assumption: $\Delta < 0.5$

Chance constraint

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

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 complex integral over **multivariate** Gaussian.

1

A set of individual chance constraints.

- Each involves one hard constraint, over a **univariate** Gaussian distribution.

2

A set of deterministic state constraints.

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

Use Boole's inequality (union bound)

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

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** constraint **over the planning horizon.**

is implied by:

Individual chance constraints

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

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

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

Variable

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

Risk allocation:

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

Decomposition of Joint Chance Constraint

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

s.t.

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

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

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

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

Decomposition of Joint Chance Constraint

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

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

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

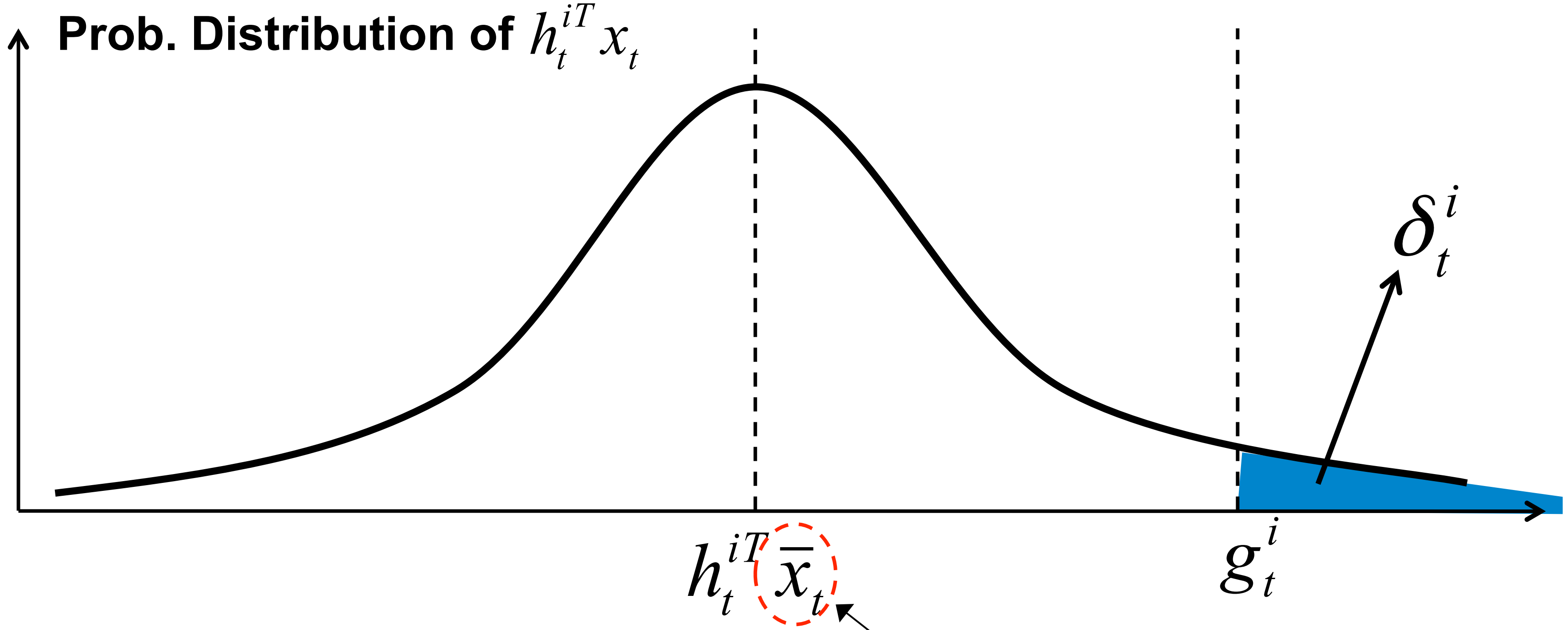
$$\sum_{t,i} \delta_t^i \le \Delta$$

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



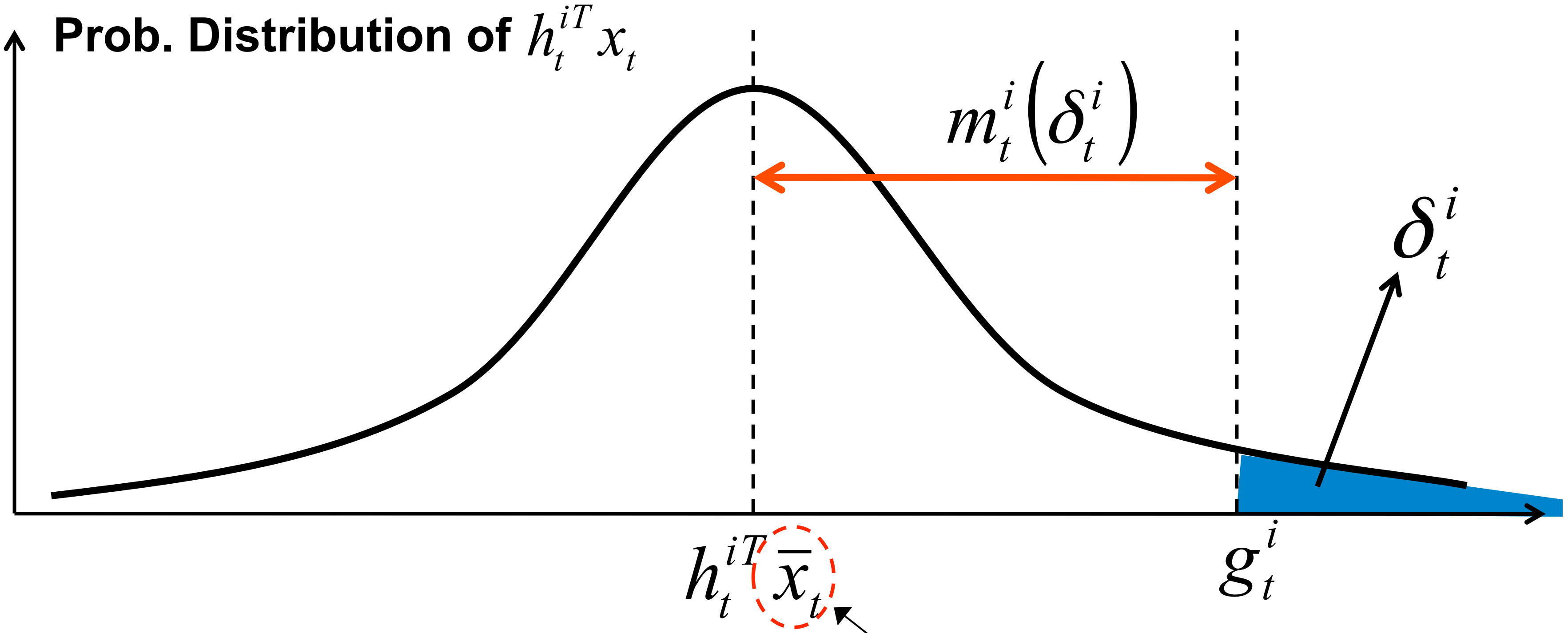
Mean state (deterministic variable)

2

Conversion to Deterministic Constraint

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

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



Mean state (deterministic variable)

2

Conversion to Deterministic Constraint

Chance constraint

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

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

Chance constraint

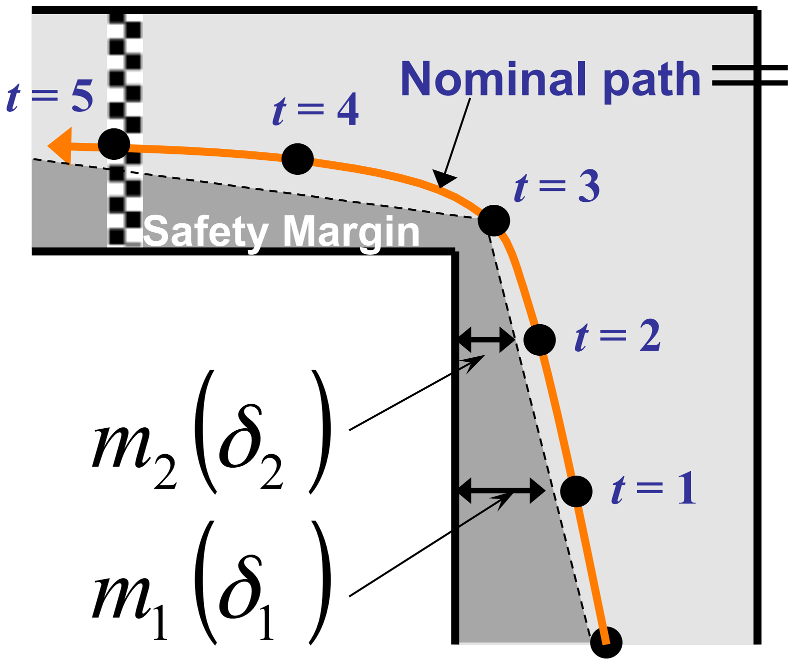
$$\Pr\left[h_t^{iT} x_t \le g_t^i\right] \ge 1-\delta_t^i$$

Deterministic constraint

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

Safety Margin

Nominal path = Mean state



[Charnes et. al. 1959]

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

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

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

2

Conversion to Deterministic Constraint

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

s.t.

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

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

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

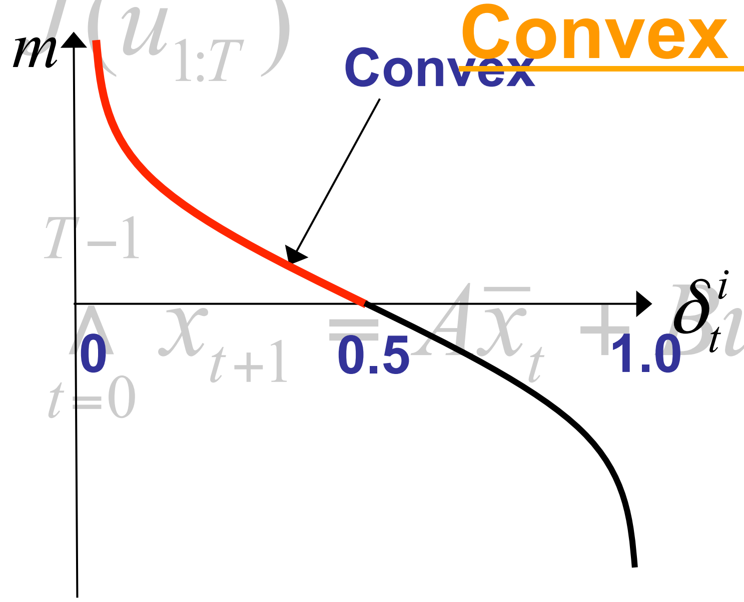
Conversion to Deterministic Constraint

Convex optimization

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

s.t.

$$\bigwedge_{t=0}^{T-1} \bar{x}_{t+1} = A\bar{x}_t + Bu_t$$



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

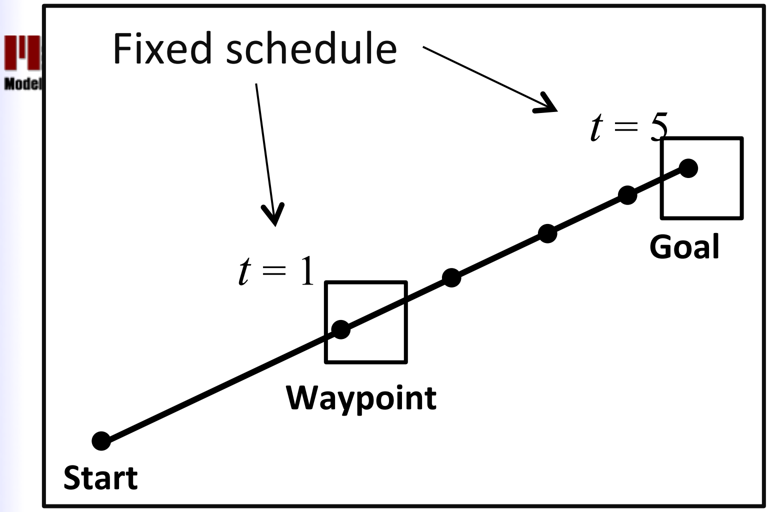
Convex if $\delta < 0.5$

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

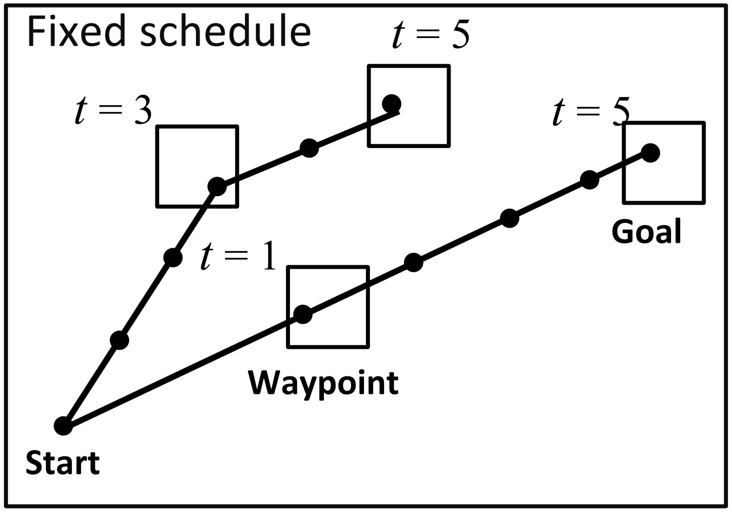
Outline

- Goal-directed, Model-Predictive Control
- Stochastic Optimization
- Iterative Risk Allocation
- Optimal Risk Allocation
  - Stochastic Linear Programs
  - Disjunctive Linear Programs
  - Probabilistic Sulu
- Multi-agent Risk Allocation

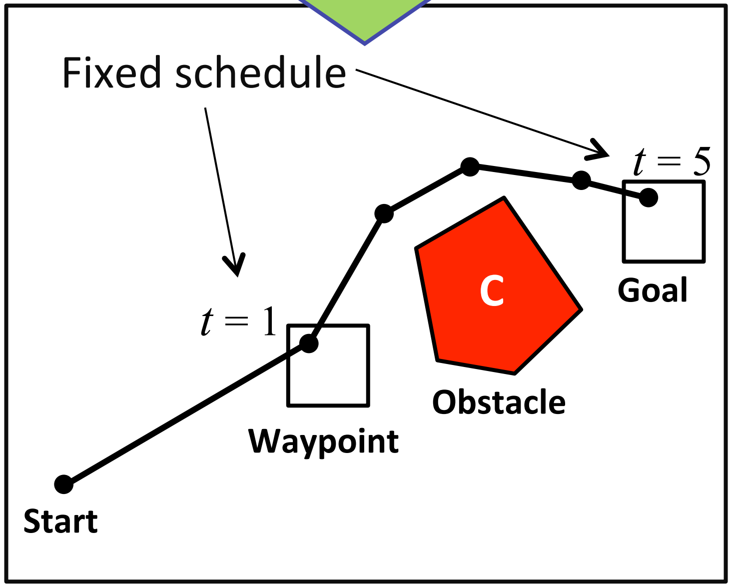
Problems



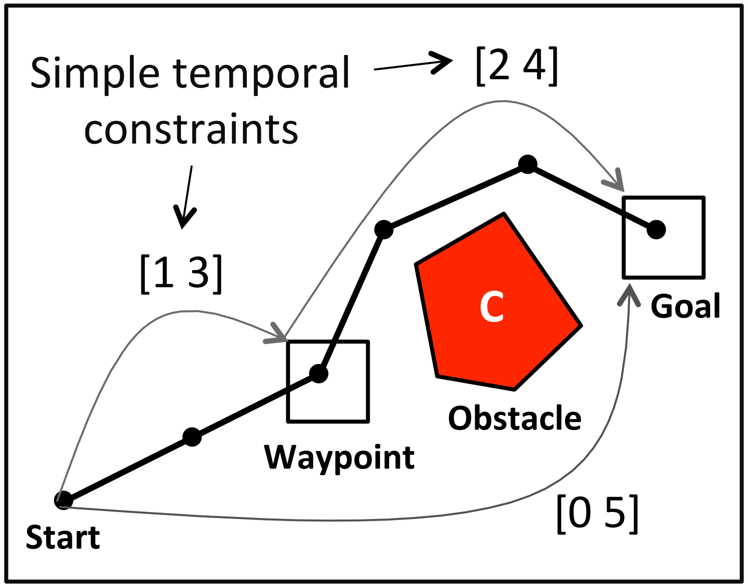
Convex, single agent



Convex, multi-agent

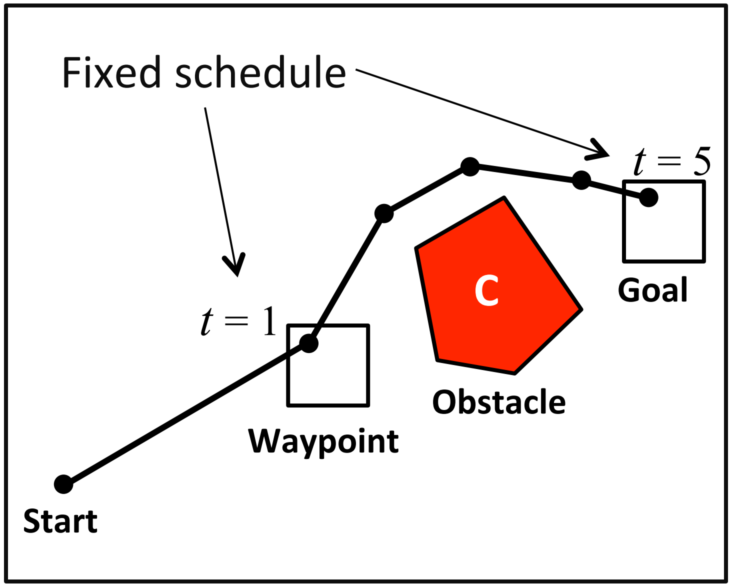


Non-convex, single agent



Non-convex, flexible schedule, single agent

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

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 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

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 Program

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

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

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

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 Program

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

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

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

Decomposition Through Risk Selection

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

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

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

Decomposition Through Risk Selection

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

Risk allocation **1**

$$\bigwedge_{t=1}^{T}\bigwedge_{n=1}^{N}\Pr\left[\underline{\bigvee_{k=1}^{K}} h_t^{n,kT} x_t \le g_t^{n,k}\right] \ge 1-\delta_t^n, \qquad \sum_{t=1,n=1}^{T,N}\delta_t^n \le \Delta$$

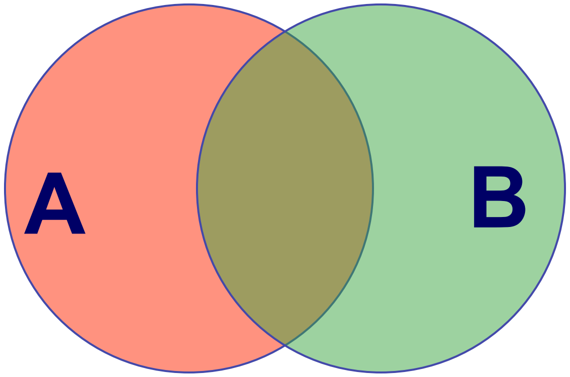
Risk selection

$$\bigwedge_{t=1}^{T}\bigwedge_{n=1}^{N}\underline{\bigvee_{k=1}^{K}}\Pr\left[h_t^{n,kT} x_t \le g_t^{n,k}\right] \ge 1-\delta_t^n, \qquad \sum_{t=1,n=1}^{T,N}\delta_t^n \le \Delta$$

2

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

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



$$\Pr[A \vee B] \ge \Pr[A]$$

$$\Pr[A \vee B] \ge \Pr[B]$$

$$\therefore \left(\Pr[A] \ge 1-\Delta \vee \Pr[B] \ge 1-\Delta\right)$$

$$\Rightarrow \Pr[A \vee B] \ge 1-\Delta$$

Solution: Branch and Bound for a Convex Disjunctive Program

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

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

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

Convex Disjunctive Programming

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

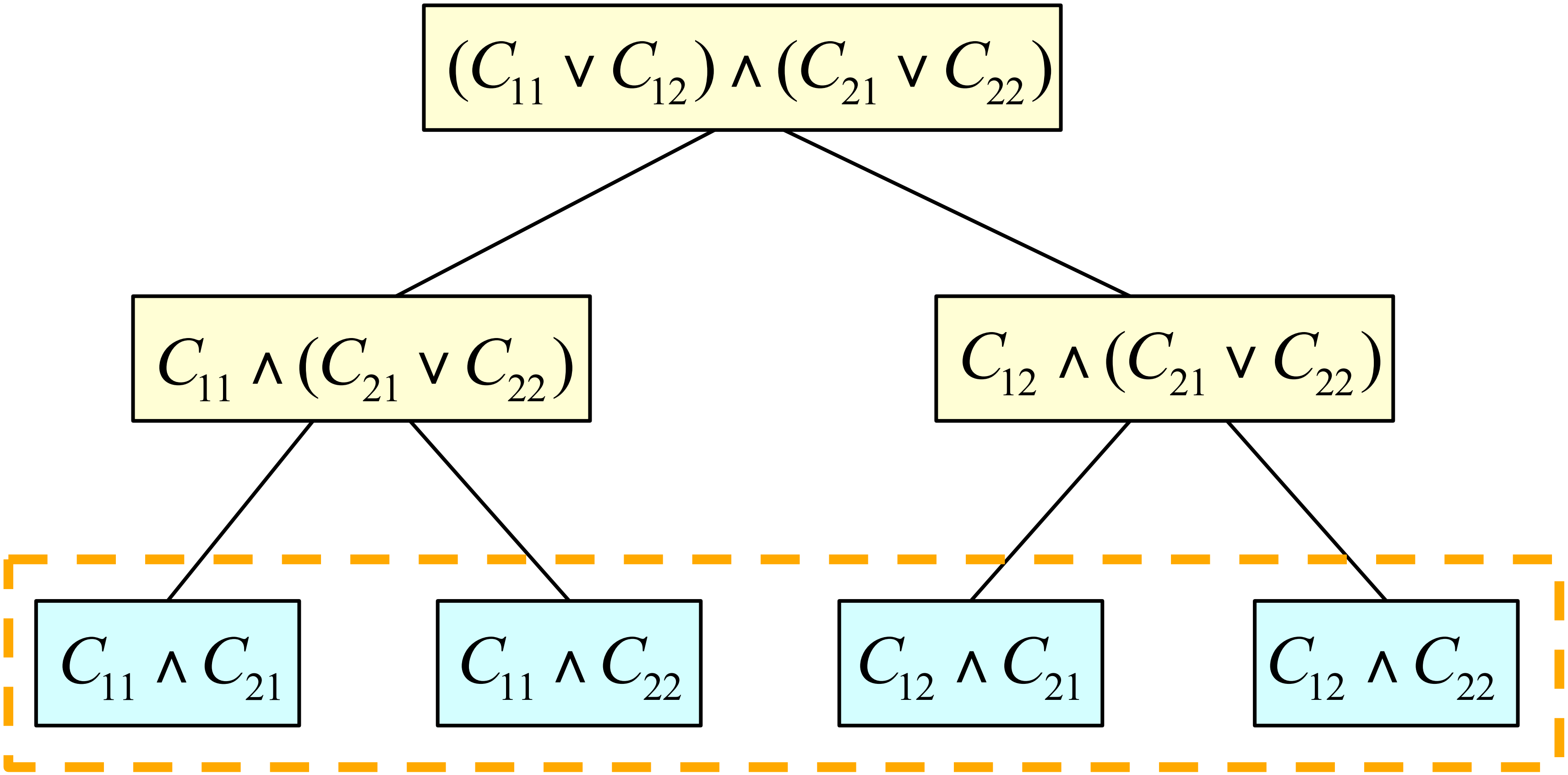
$$s.t. \quad \bigwedge_{t=1}^{T} \bigwedge_{n=1}^{N} \bigvee_{k=1}^{K} h_t^{n,kT} \bar{x}_t \leq g_t^{n,k} - m_t^{n,k}\left(\delta_t^n\right)$$

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

Stochastic DLP Branch and Bound

Repeat until no clauses left:

1. **Select clause.**
2. **Split on disjuncts.**



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

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

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

$C_{11} \wedge C_{21}$

$C_{11} \wedge C_{22}$

$C_{12} \wedge C_{21}$

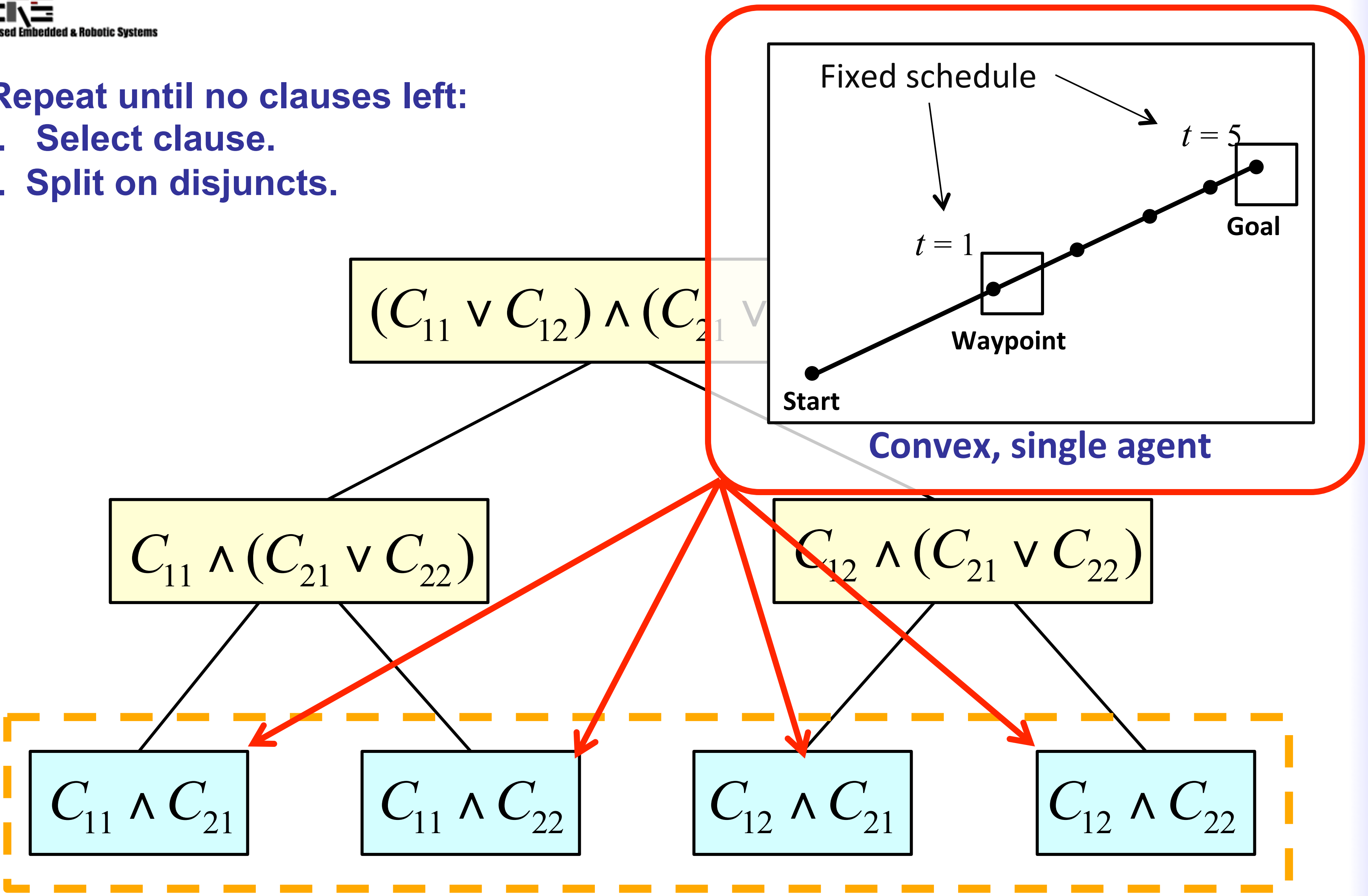
$C_{12} \wedge C_{22}$

Convex Optimization Problems

Stochastic DLP Branch and Bound

Repeat until no clauses left:

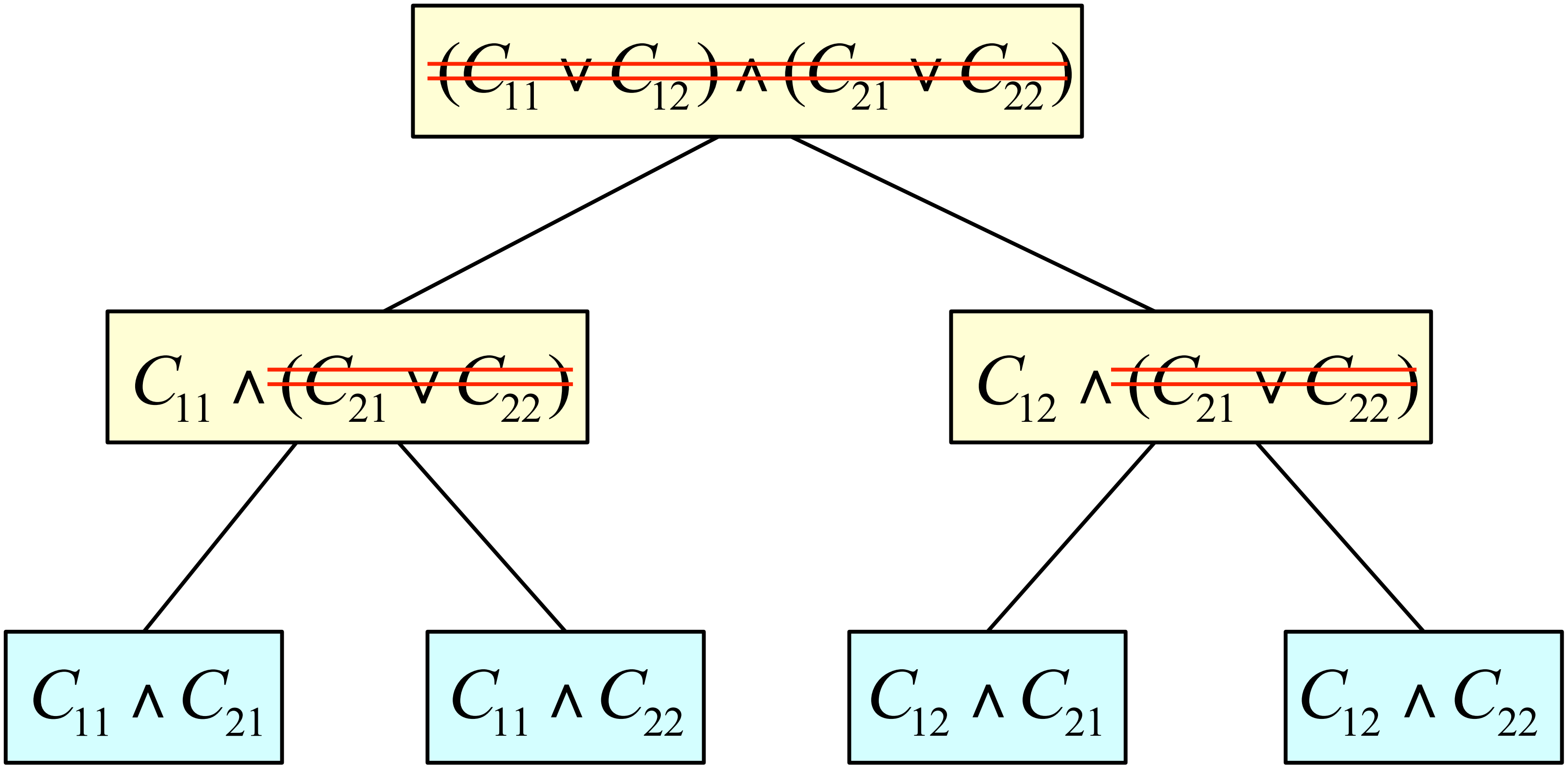
1. **Select clause.**
2. **Split on disjuncts.**



Convex Optimization

Bound Through Convex Relaxation

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



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

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

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

$C_{11} \wedge C_{21}$

$C_{11} \wedge C_{22}$

$C_{12} \wedge C_{21}$

$C_{12} \wedge C_{22}$

Subproblems of BnB (non-linear)

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

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

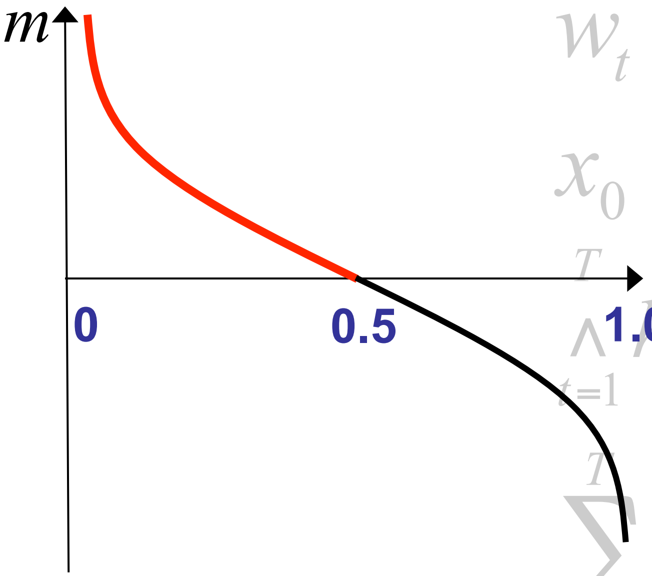
$$w_t \sim N(0,\Sigma_t)$$

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

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

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

Nonlinear



Subproblems of BnB (non-linear)

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

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

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

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

Nonlinear

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

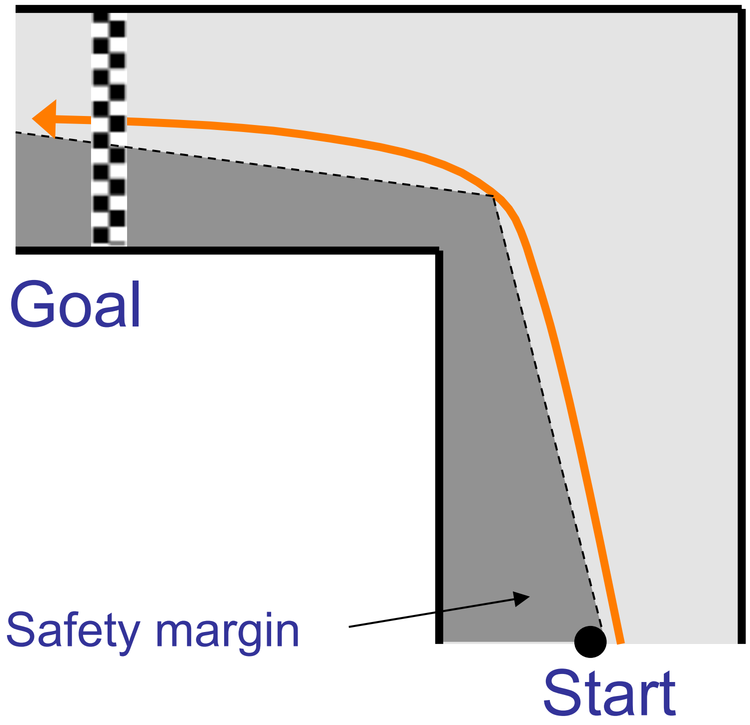
$$\cancel{\sum_{t=1}^{T} \delta_t \le \Delta,} \; \delta_t \ge 0$$

Fixed Risk Relaxation

$$\delta_t = \Delta$$

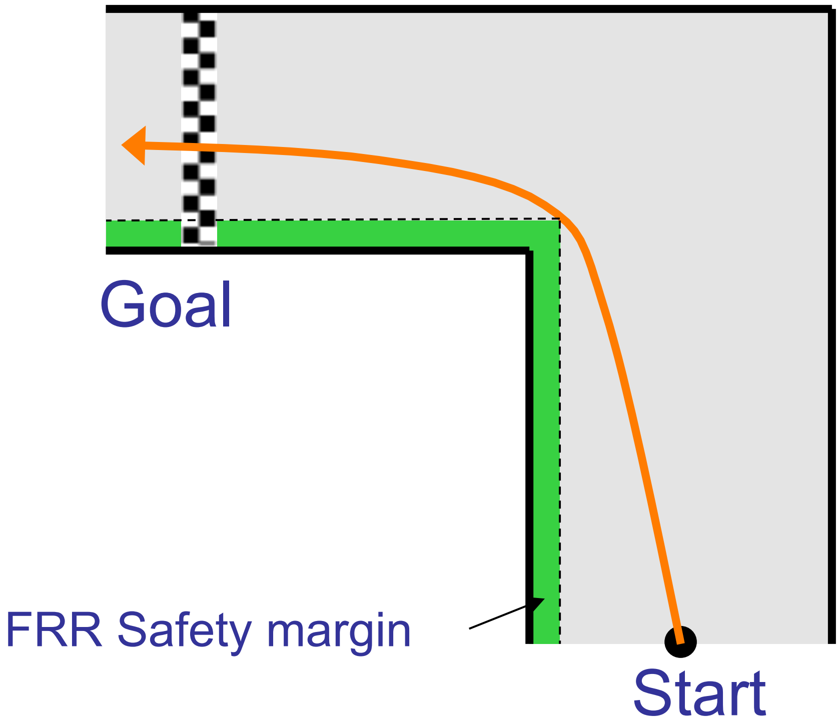
Fixed Risk Relaxation: Intuition

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** for the cost of the original problem.

Approach: ***Fixed Risk Relaxation*** (**FRR**)

- **FRR**: linear relaxation of each subproblem.
  - Has only linear constraints (**typically LP / QP**).
  - Gives **lower bound** on the cost of sub-problem.
  - May generate infeasible solution to original problem.

Fixed Risk Relaxation (Linear)

$$
\begin{aligned}
&\min_{u_{1:T}} J(u_{1:T}) \\
&s.t. \quad \forall_{0\le t\le T-1} \; x_{t+1} = Ax_t + Bu_t + w_t \\
&\qquad w_t \sim N(0,\Sigma_t) \\
&\qquad x_0 \sim N(\bar{x}_0,\Sigma_{x,0}) \\
&\qquad \bigwedge_{t=1}^{T} h_t^{i(t)T}\bar{x}_t \le g_t^{i(t)} - m_t^{i(t)}(\Delta)
\end{aligned}
$$

Constant

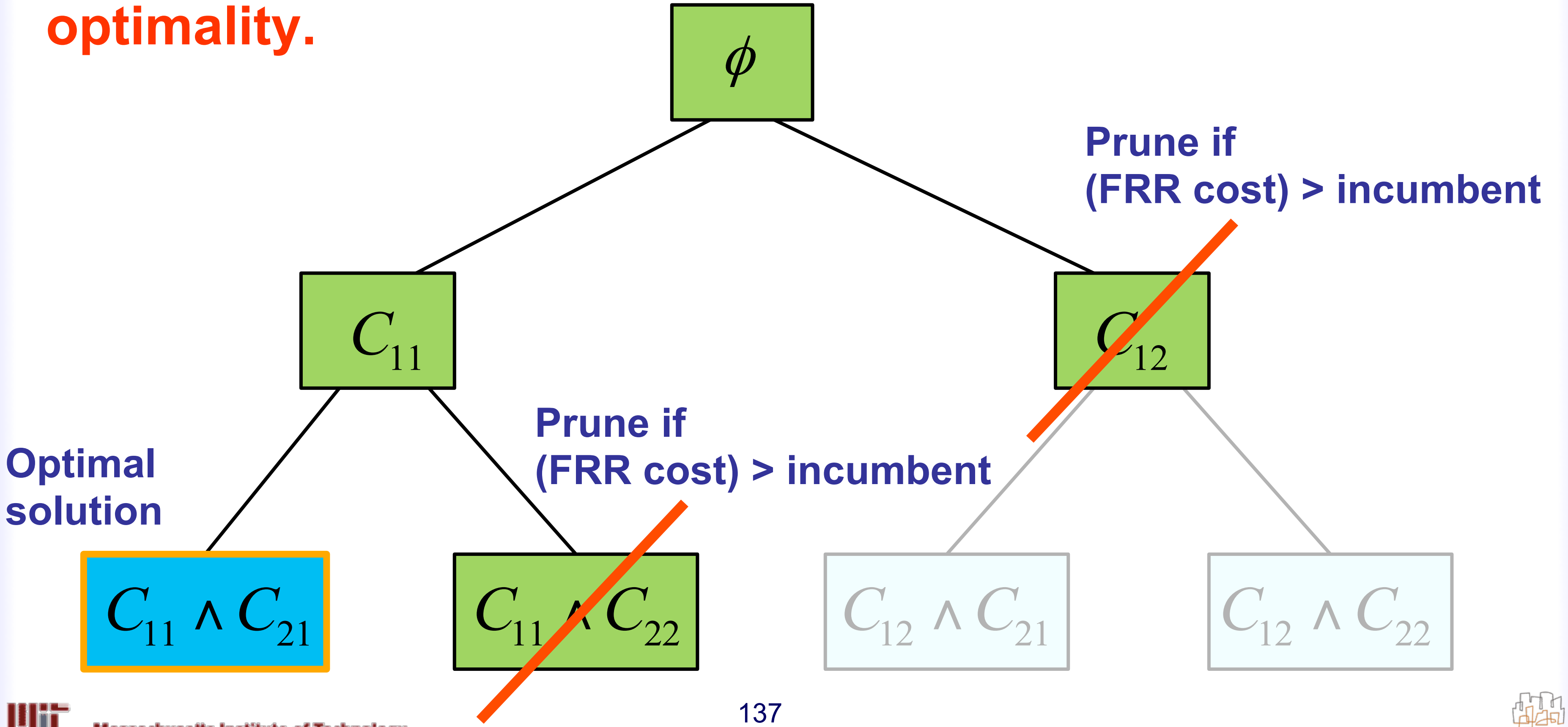
Fixed Risk Relaxation

$$\delta_t = \Delta$$

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

Algorithm: BnB + **FRRs**

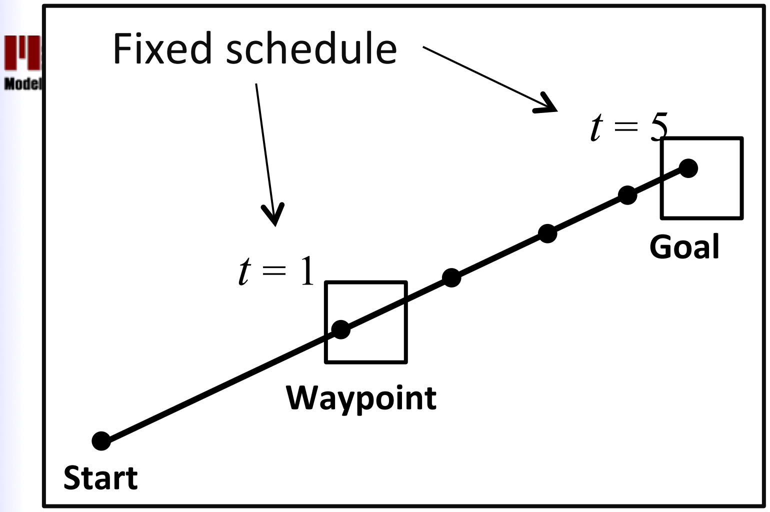
- Solve **FRR**s of subproblems to reduce computation time.
- Solve subproblem ***without relaxation*** at unpruned leaf nodes to obtain exact solution.
- Significantly reduces computation time **without compromising optimality.**



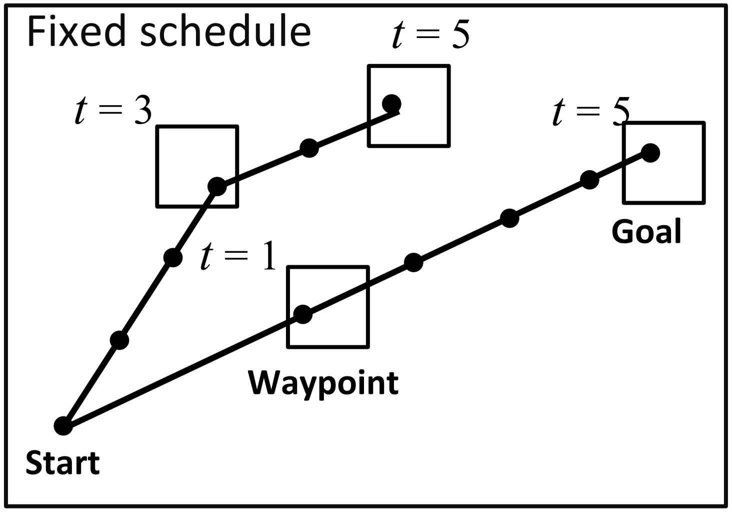
Outline

- Goal-directed, Model-Predictive Control
- Stochastic Optimization
- Iterative Risk Allocation
- Optimal Risk Allocation
  - Stochastic Linear Programs
  - Disjunctive Linear Programs
  - Probabilistic Sulu
- Appendix: Multi-agent Risk Allocation

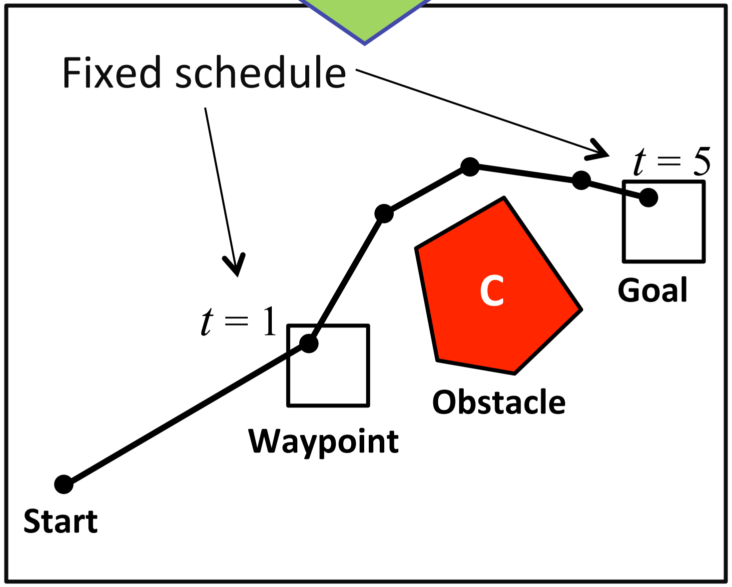
Problems



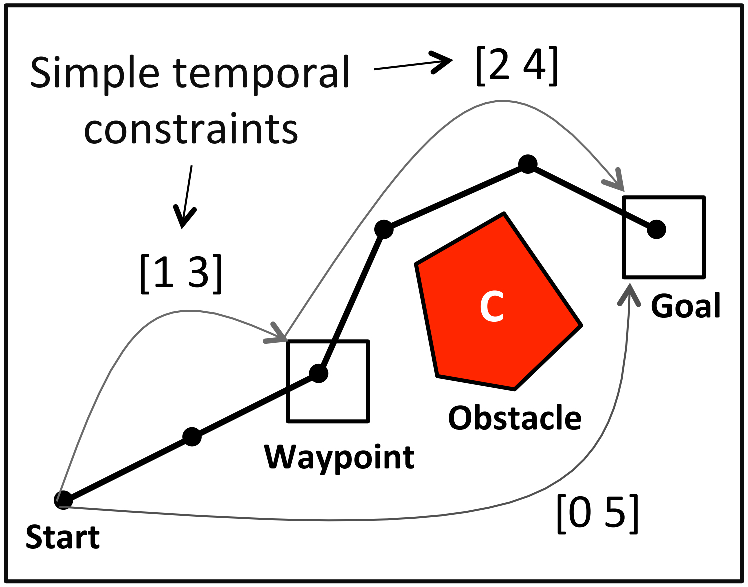
Convex, single agent



Convex, multi-agent



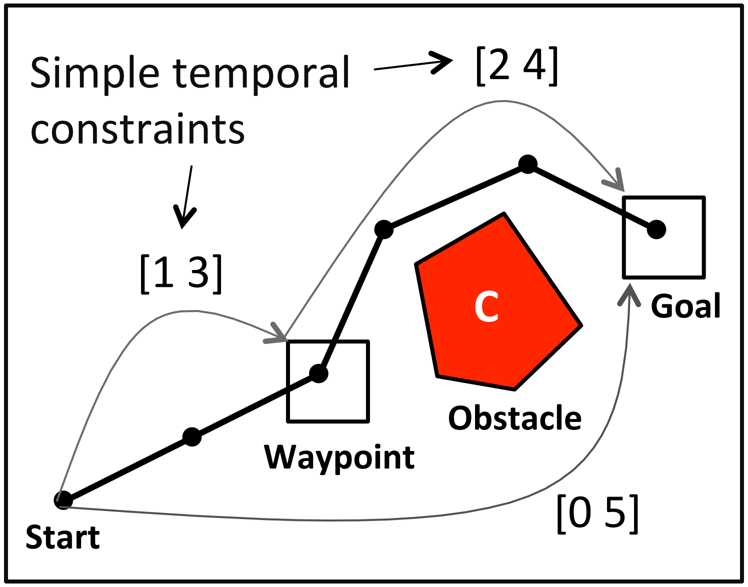
Non-convex, single agent



Non-convex, flexible schedule, single agent

Problem Formulation

Non-convex, flexible schedule



$$\min_{\boldsymbol{u}_{1:N}\in\mathbb{U}^N, \boxed{s\in\mathcal{S}_F}} \quad J(\boldsymbol{u}_{1:N}, \bar{\boldsymbol{x}}_{1:N}, \boxed{s})$$

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

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

Flexible schedule

Two-layer Approach

Outer-loop: Schedule optimization

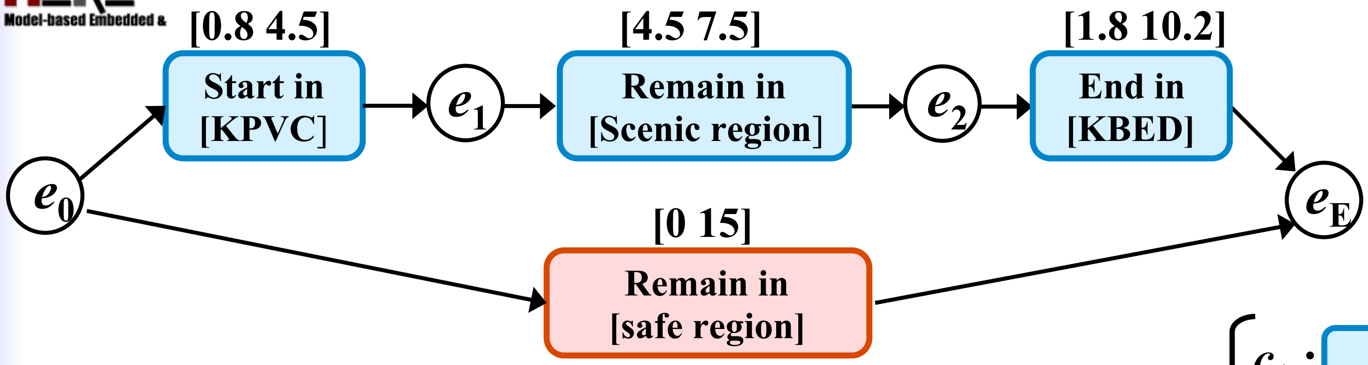
```
function p-Sulu(ccqsp)
  incumbent ← INF;
  for s $\in\mathcal{S}_F$
    (J*, U*) ← innerLoop(s,ccqsp);
    if J* < incumbent
      incumbent ← J*;
      solution ← (s,U*)
    endif
  endfor
 return solution;
```

Inner-loop: fixed schedule CC-QSP as a Stochastic-DLP

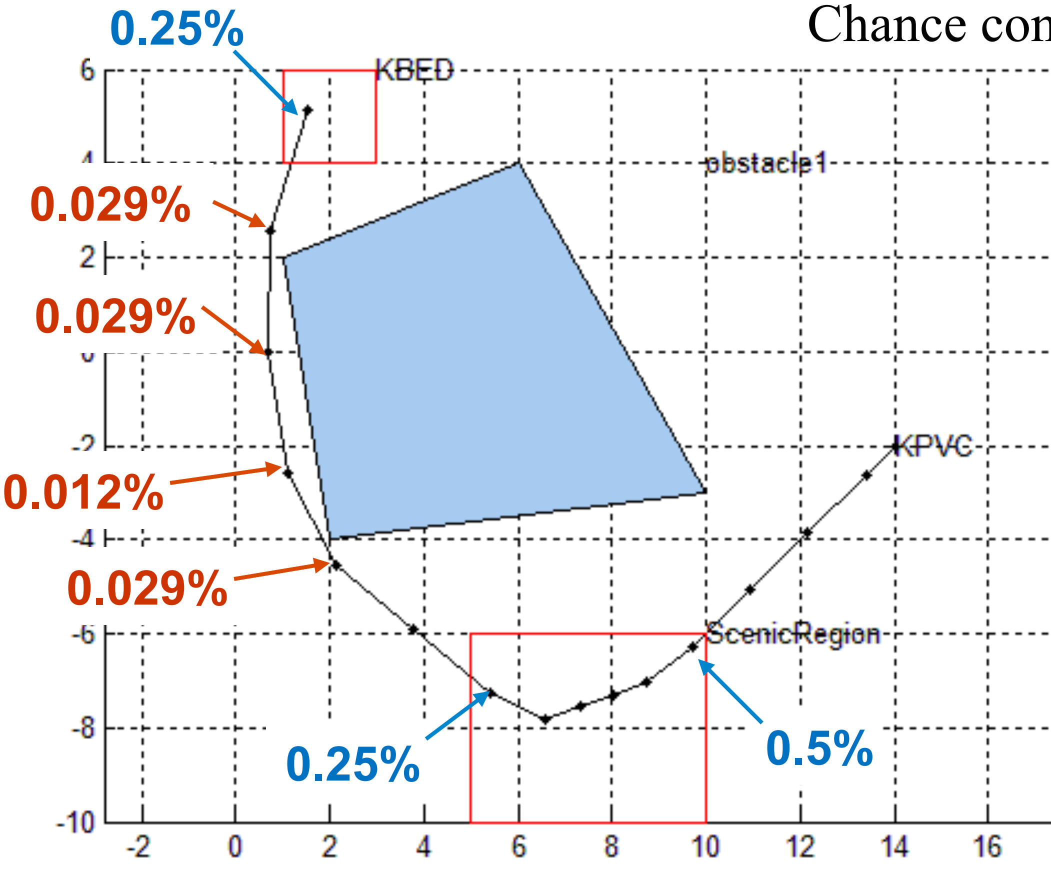
```
function innerLoop(s,ccqsp)
  Solve chance-constrained optimal control with s and ccqsp;
  U* ← Optimal control sequence;
  J* ← Optimal objective value;
  return (J*, U*);
```

NIRA Algorithm

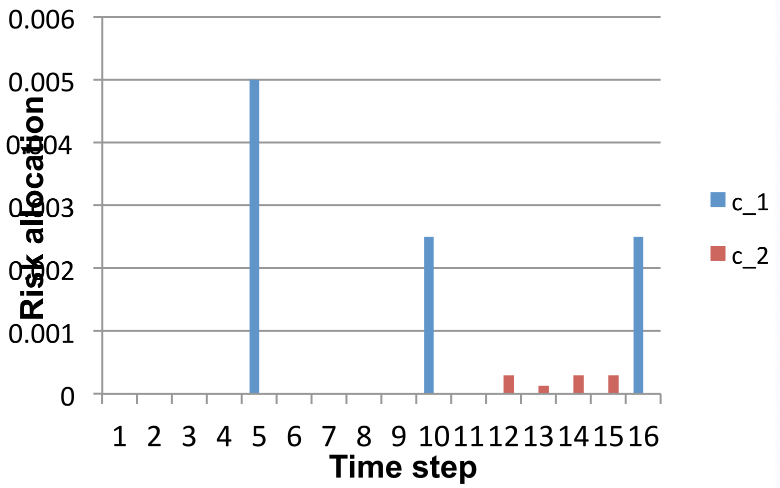
Results 1: Personal Transport Scenario



Chance constraints: $\begin{cases} c_1 : 🟦 & \cdots \delta = 0.01 \\ c_2 : 🟥 & \cdots \delta = 0.001 \end{cases}$

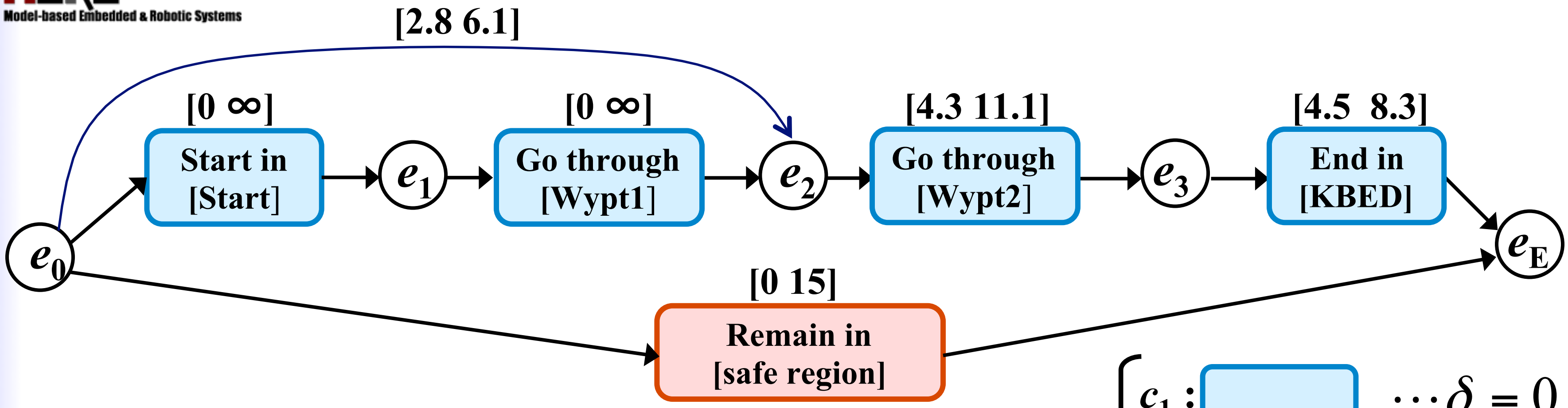


Computation time: 7.06 sec



Results 2: 2 Obstacles, 3 Goals

Model-based Embedded & Robotic Systems



[2.8 6.1]

e_0

[0 ∞]

Start in [Start]

e_1

[0 ∞]

Go through [Wypt1]

e_2

[4.3 11.1]

Go through [Wypt2]

e_3

[4.5 8.3]

End in [KBED]

e_E

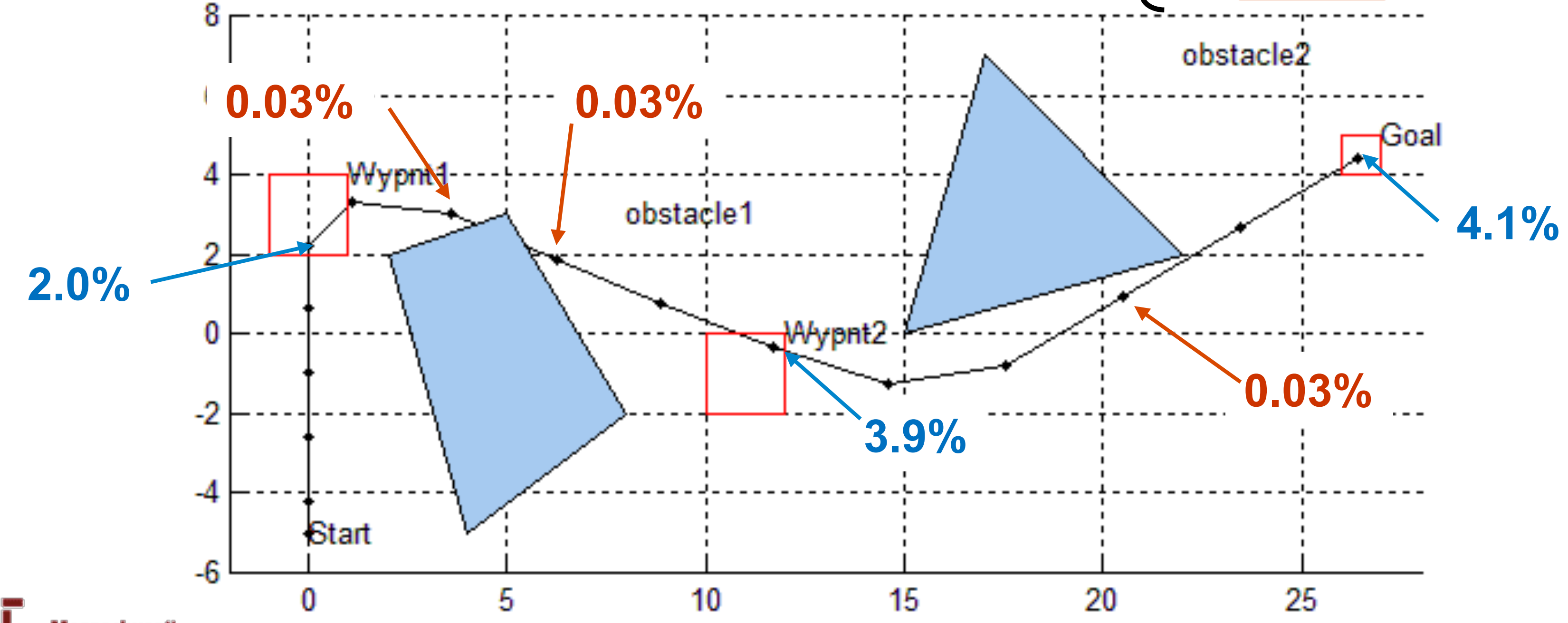
[0 15]

Remain in [safe region]

Computation time: 8544 sec (2hr 23 min) Chance constraints:

c_1 : 🟦 $\cdots \delta = 0.1$

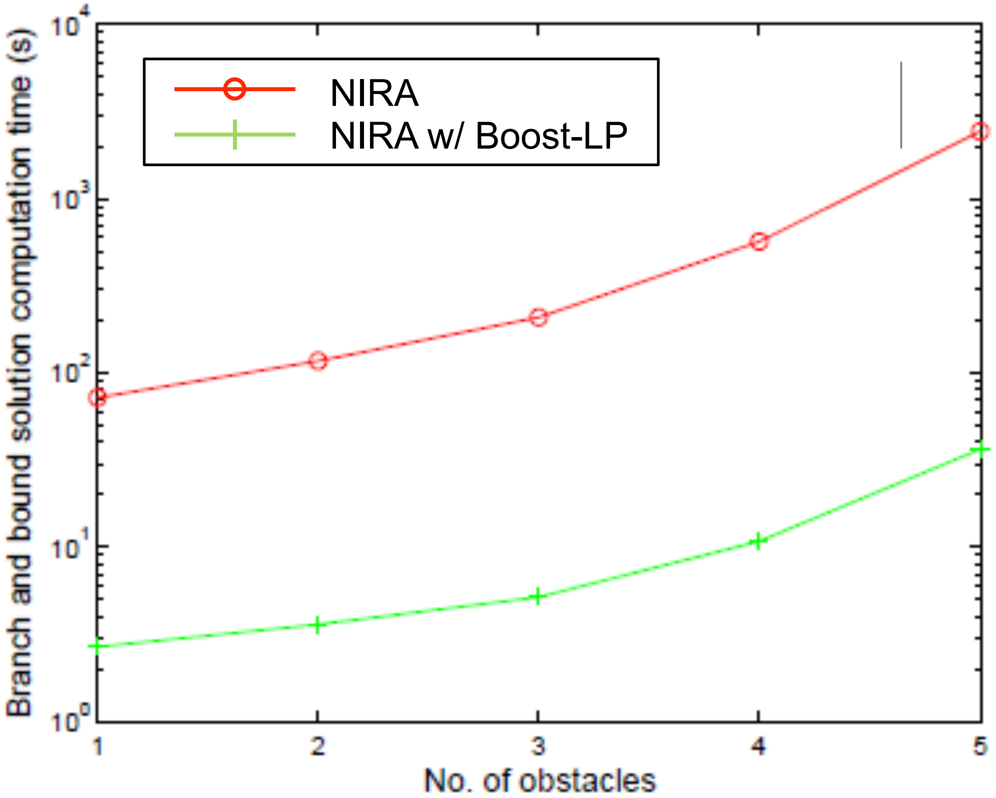
c_2 : 🟥 $\cdots \delta = 0.001$



Performance Improvement Using Boosting Tree-based Regression

Scenario #	1	2	3	4
NIRA	135.21	219.76	79.99	80.15
NIRA w/ Boost-LP	3.84	4.15	3.03	2.93

- **Both algorithms always result in the same solution**



T=20, Δ=0.01

Scenaros:

#1: 2 obstacles and no waypoint

#2: 2 obstacles and 2 waypoints

#3: 1 obstacle and 1 waypoint, trained with different disturbance level

#4: 1 obstacle with 1 waypoint, trained with different control constraints

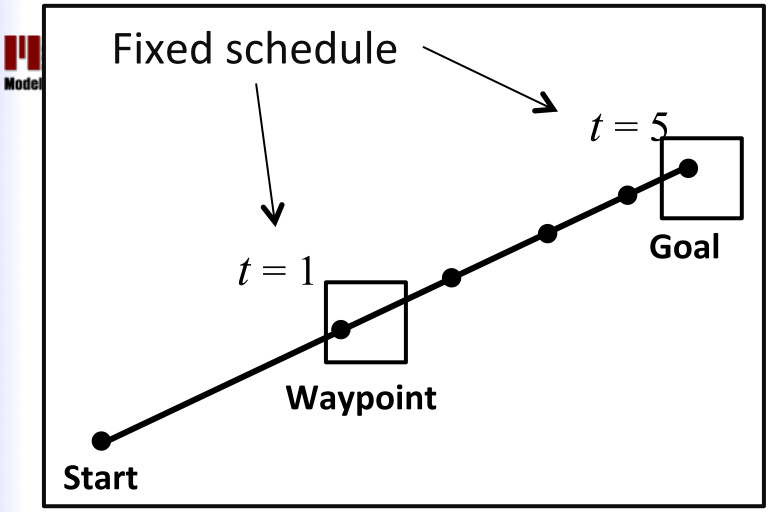
*** Banerjee, A. G., & Roy, N. (2010). Learning Solutions of Similar Linear Programming Problems using Boosting Trees. CSAIL technical report MIT-CSAIL-TR-2010-045**

P-Sulu Performing Rendezvous and Docking on Spheres

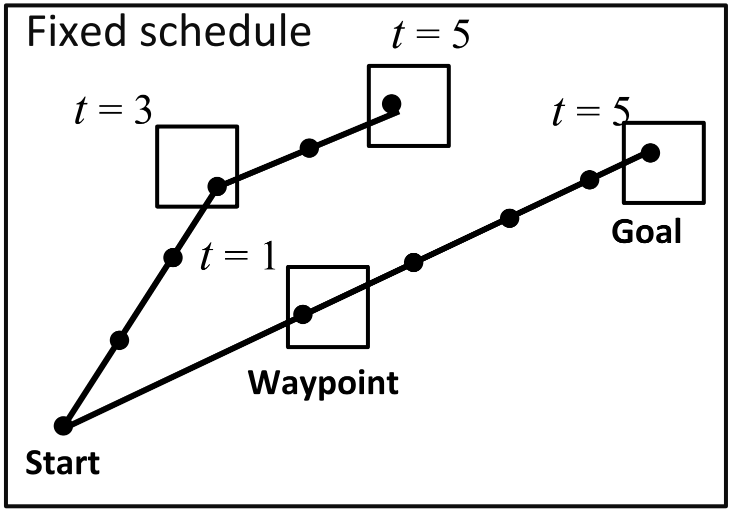
Model-based Embedded & Robotic Systems

Massachusetts Institute of Technology

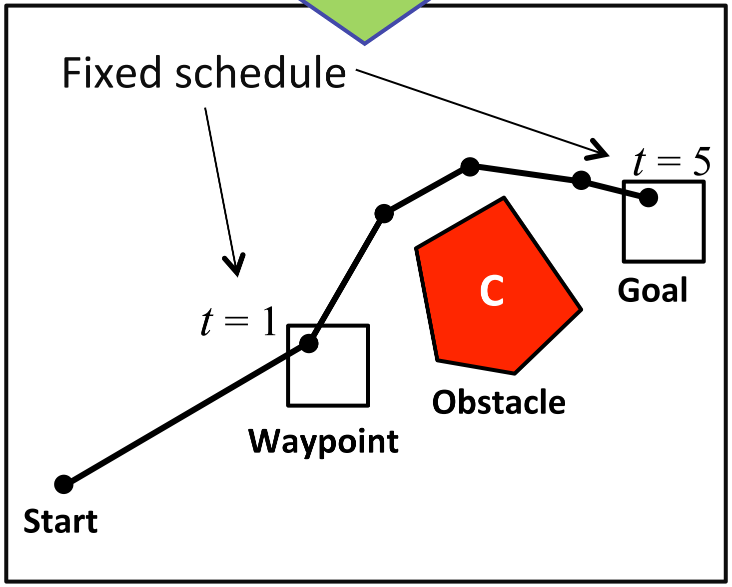
Problems



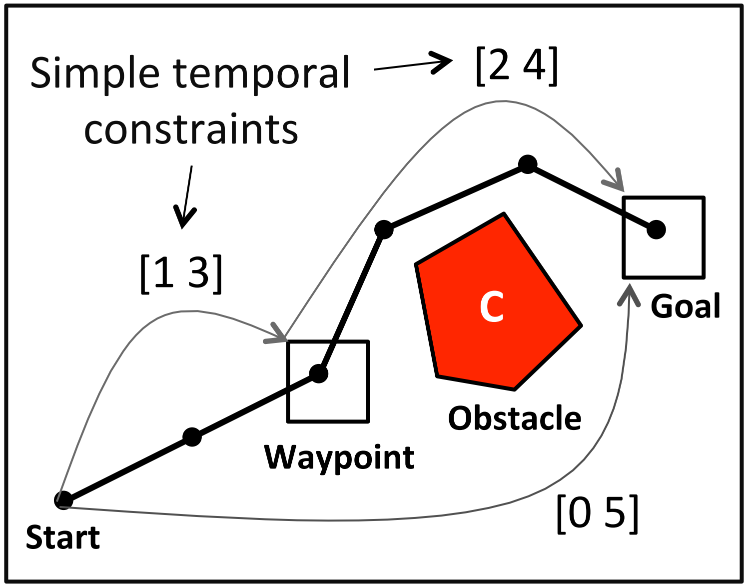
Convex, single agent



Convex, multi-agent

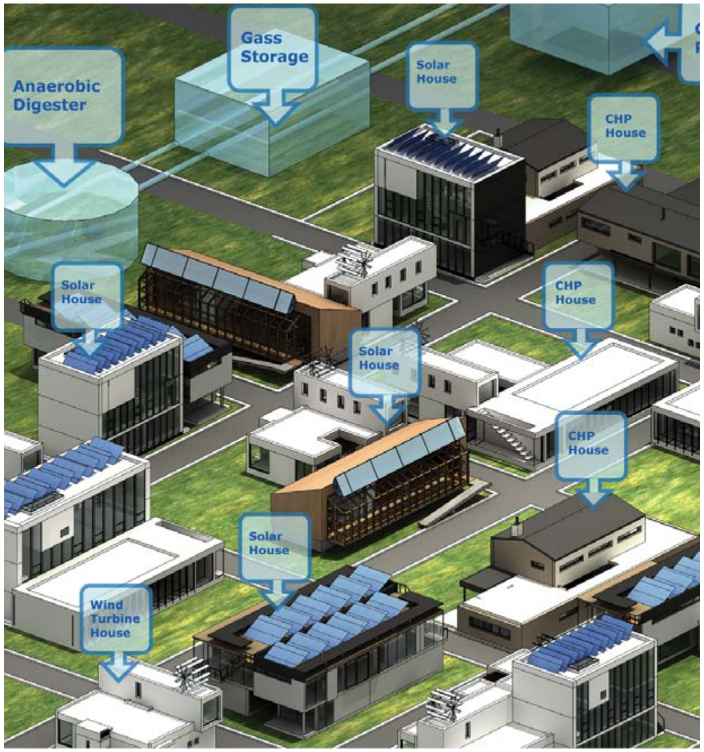


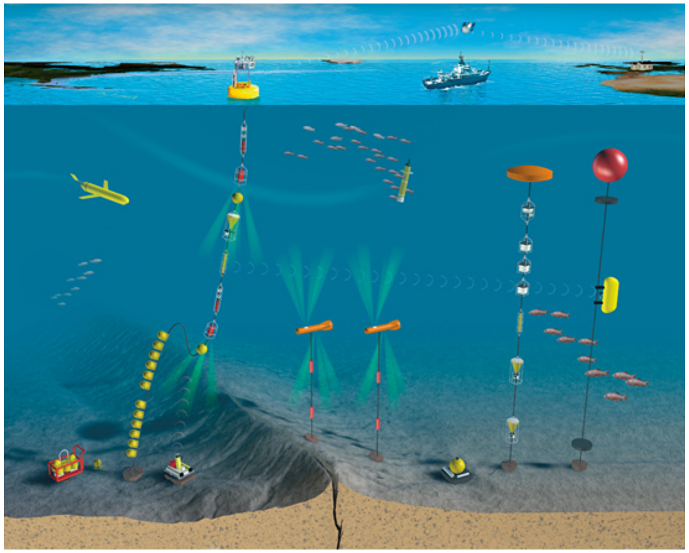
Non-convex, single agent



Non-convex, flexible schedule, single agent

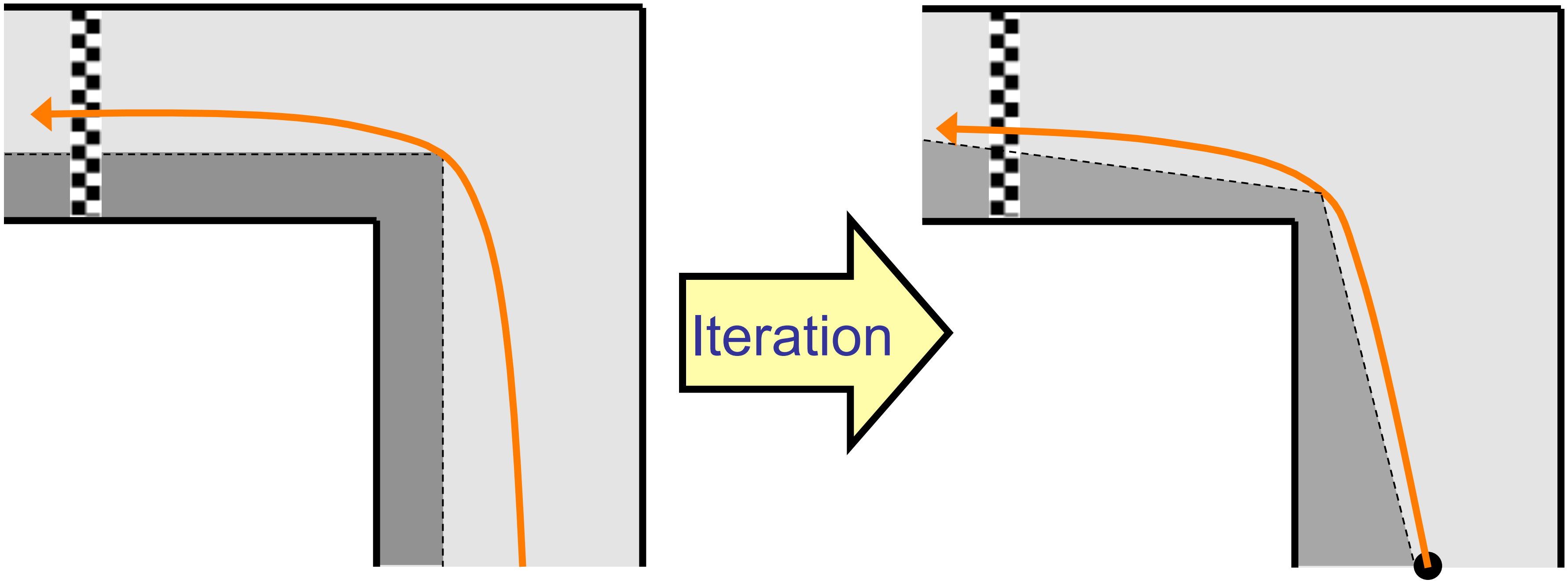
Facilitating Sustainability Requires Managing Risk





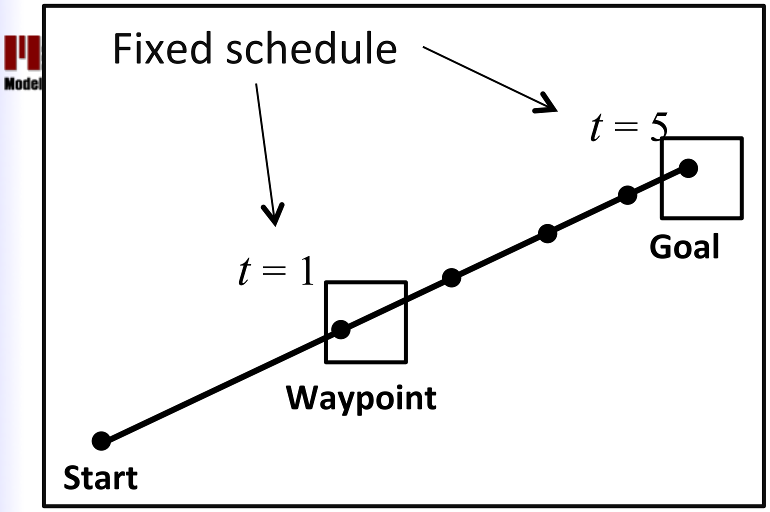
Risk Allocation

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

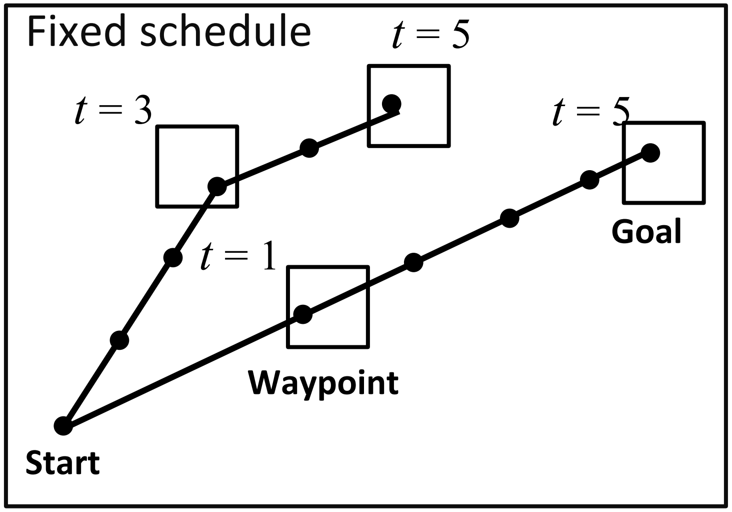


1. IRA: reallocates risk manually.
2. CRA,NRA: standard solver reallocates risk.

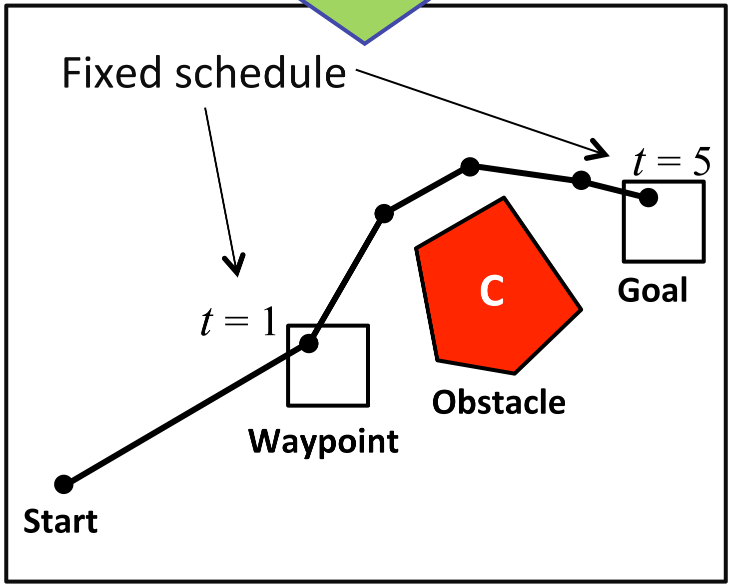
Risk-bounded Planning



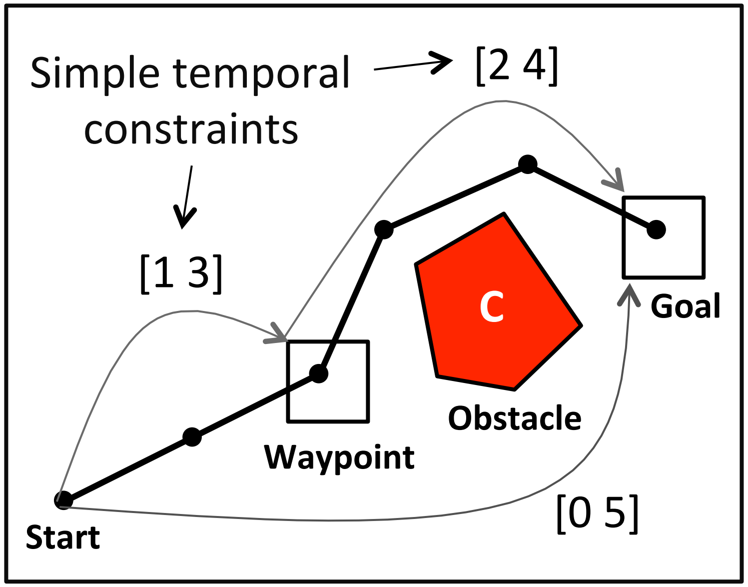
Convex, single agent



Convex, multi-agent

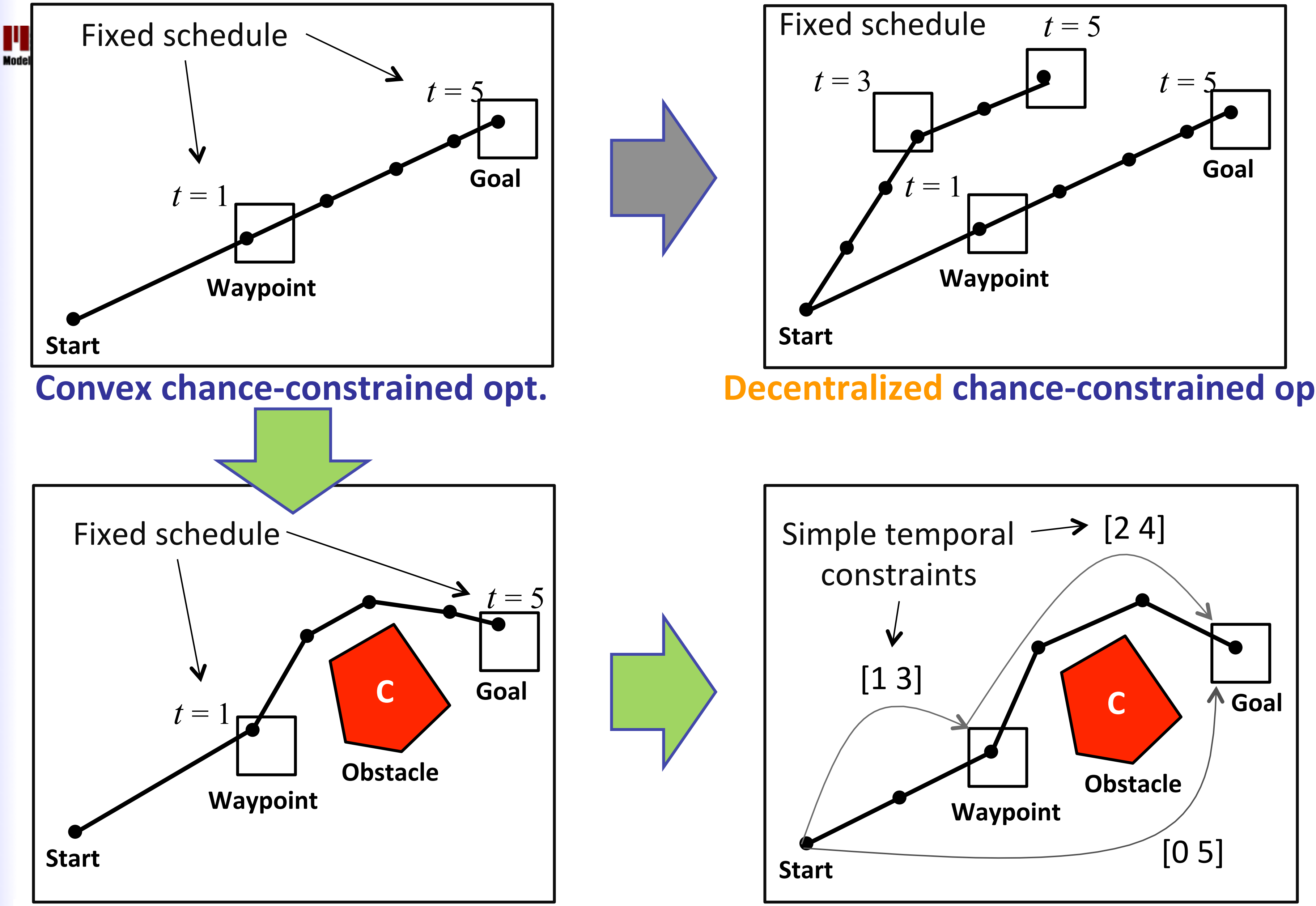


Non-convex, single agent



Non-convex, flexible schedule, single agent

Optimization Problems



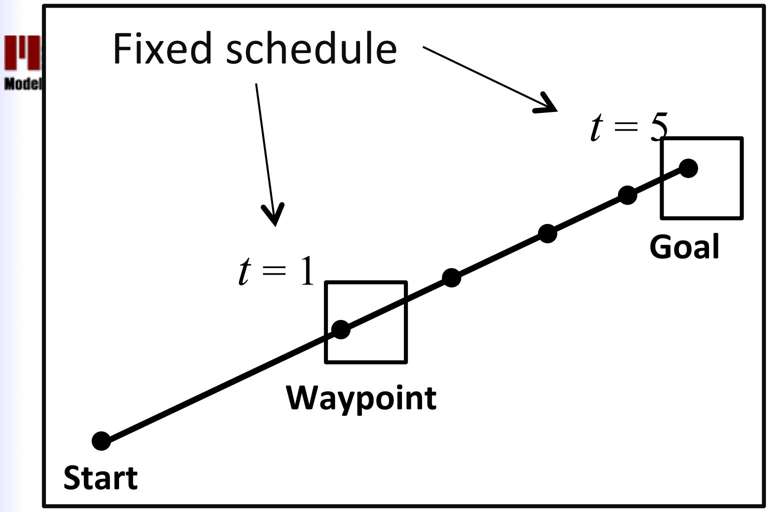
Convex chance-constrained opt.

Decentralized chance-constrained opt

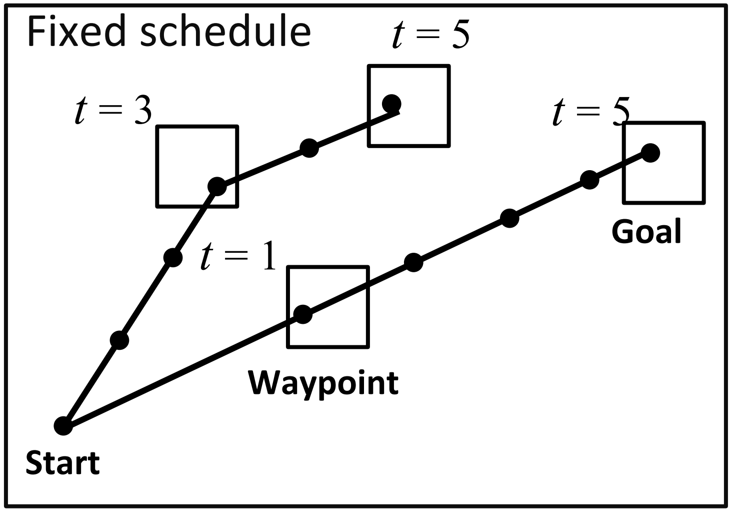
Non-convex, chance-constrained opt

Chance-constrained planning & scheduling

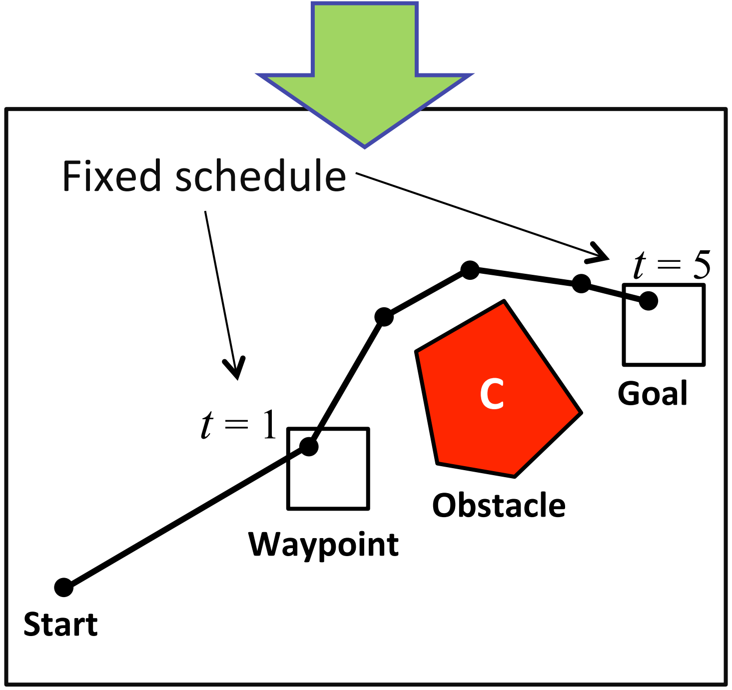
Risk Allocation Algorithms



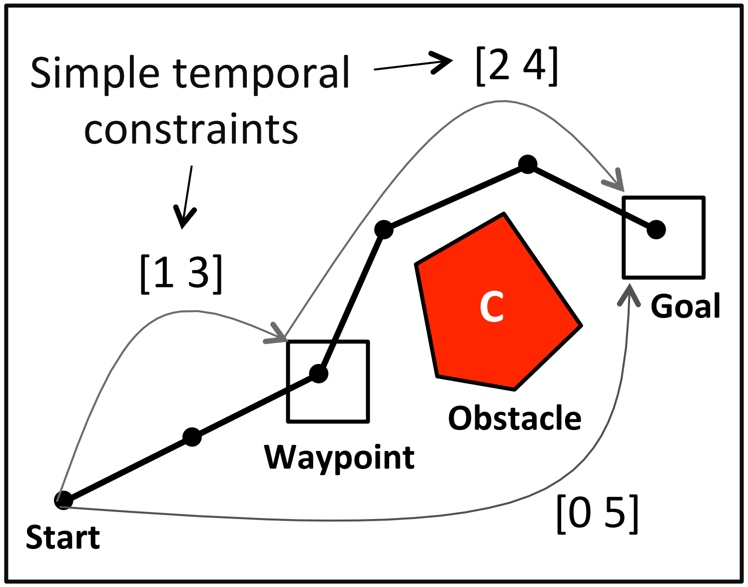
IRA (Iterative Risk Allocation)



MIRA (Market-based IRA)



IRA (Non-convex Risk Allocation)

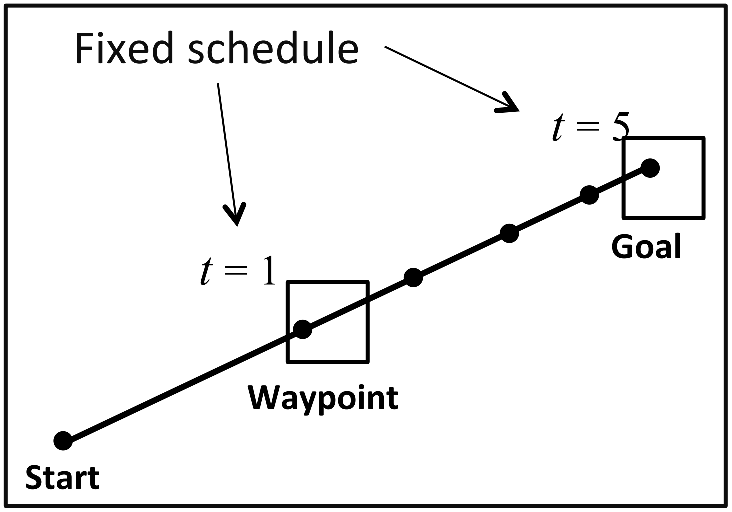


p-Sulu (probabilistic Sulu)

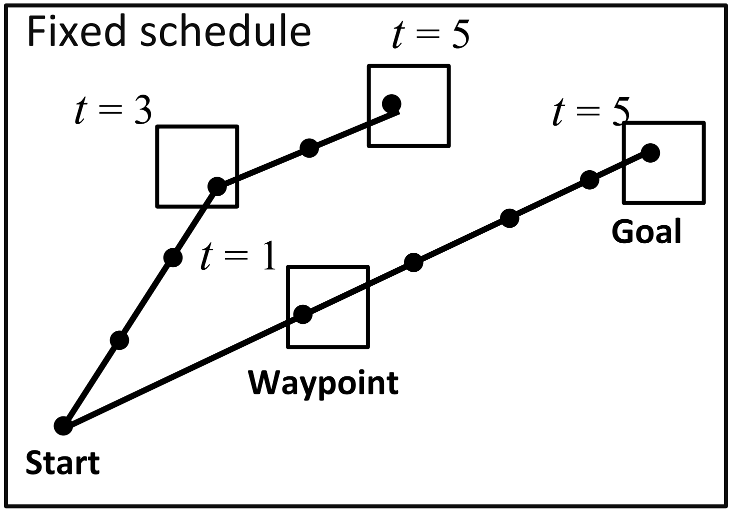
Outline

- Model-Predictive Control
- Stochastic Optimization
- Iterative Risk Allocation
- Optimal Risk Allocation
- Appendix: Multi-agent Risk Allocation

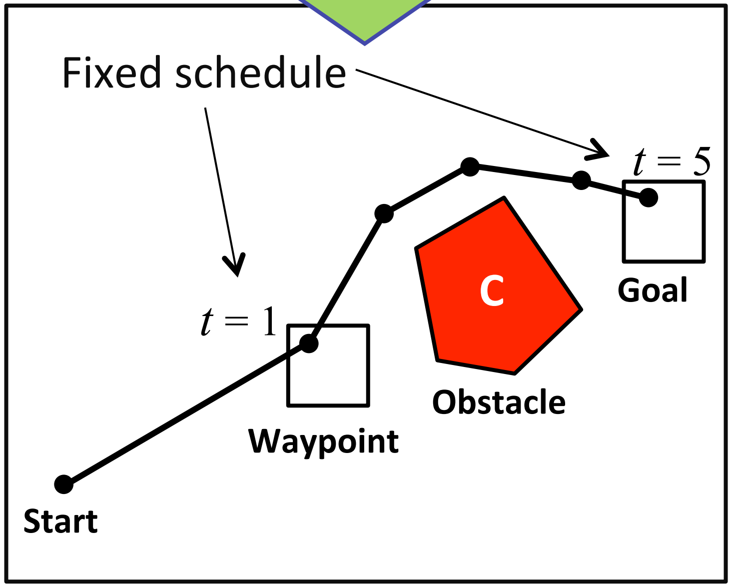
Algorithms



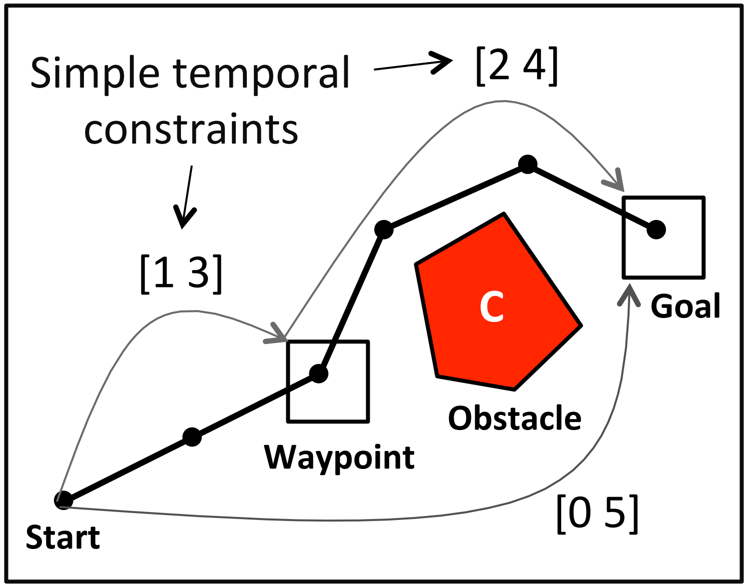
IRA (Iterative Risk Allocation)



MIRA (Market-based IRA)



IRA (Non-convex Risk Allocation)



p-Sulu (probabilistic Sulu)

Problem Formulation for Multi-agent

Single agent

$$\min_{U \in \mathbf{U}} J(U)$$

$$s.t. \quad \Pr\left[\bigwedge_{n=1}^{N} h_n^T X \le g_n^i \right] \ge 1 - \Delta$$

Multi-agent

$$\min_{U^{1:I} \in \mathbf{U}^{1:I}} \sum_{i=1}^{I} J^i(U^i)$$

$$s.t. \quad \Pr\left[\bigwedge_{i=1}^{I} \bigwedge_{n=1}^{N_i} h_n^{iT} X^i \le g_n^i \right] \ge 1 - \Delta$$

***i*:** Index of agents

I agents, ***Nⁱ*** state constraints for *i* 'th agent

$$X^i \equiv \begin{pmatrix} x_1^i \\ \vdots \\ x_T^i \end{pmatrix} \quad U^i \equiv \begin{pmatrix} u_1^i \\ \vdots \\ u_T^i \end{pmatrix}$$

Problem Formulation for Multi-agent

Single agent

$$\min_{U \in \mathbf{U}} J(U)$$

$$s.t. \quad \Pr\left[\bigwedge_{n=1}^{N} h_n^T X \le g_n \right] \ge 1 - \Delta$$

Multi-agent

$$\min_{U^{1:I} \in \mathbf{U}^{1:I}} \boxed{\sum_{i=1}^{I}} J^i(U^i)$$

$$s.t. \quad \Pr\left[\bigwedge_{i=1}^{I} \bigwedge_{n=1}^{N_i} h_n^{iT} X^i \le g_n^i \right] \ge 1 - \Delta$$

***i*:** Index of agents

I agents, N^i state constraints for i'th agent

- Minimize **aggregate cost**

$$X^i \equiv \begin{pmatrix} x_1^i \\ \vdots \\ x_T^i \end{pmatrix} \quad U^i \equiv \begin{pmatrix} u_1^i \\ \vdots \\ u_T^i \end{pmatrix}$$

Problem Formulation for Multi-agent

Single agent

$$\min_{U \in \mathbf{U}} J(U)$$

$$s.t. \quad \Pr\left[\bigwedge_{n=1}^{N} h_n^T X \le g_n^i \right] \ge 1 - \Delta$$

Multi-agent

$$\min_{U^{1:I} \in \mathbf{U}^{1:I}} \sum_{i=1}^{I} J^i(U^i)$$

$$s.t. \quad \Pr\left[\boxed{\bigwedge_{i=1}^{I}} \bigwedge_{n=1}^{N_i} h_n^{iT} X^i \le g_n^i \right] \ge 1 - \Delta$$

i**:** Index of agents

I agents, ***N^i*** state constraints for *i* ’th agent

- Minimize aggregate cost
- Bound the probability that **all agents** satisfy **all constraints**
  - System fails if one agent violates constraints.

Risk Allocation between Agents

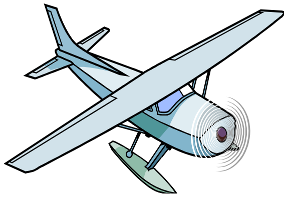
- Need to optimize **risk allocation between agents** since sensitivity to risk is different

User

specifies

System's risk bound: **0.1%**

0.02% + **0.08%**



Risk is distributed among agents

$$\sum\left(\textit{Individual risk bounds}\right) \le \textit{System's risk bound}$$

Multi-agent

$$\min_{U^{1:I}\in\mathbf{U}^{1:I}} \sum_{i=1}^{I} J^i(U^i)$$

$$s.t.\quad \Pr\left[\bigwedge_{i=1}^{I}\bigwedge_{n=1}^{N_i} h_n^{iT}X^i \le g_n^i\right] \ge 1-\Delta$$

Decomposed, deterministic form

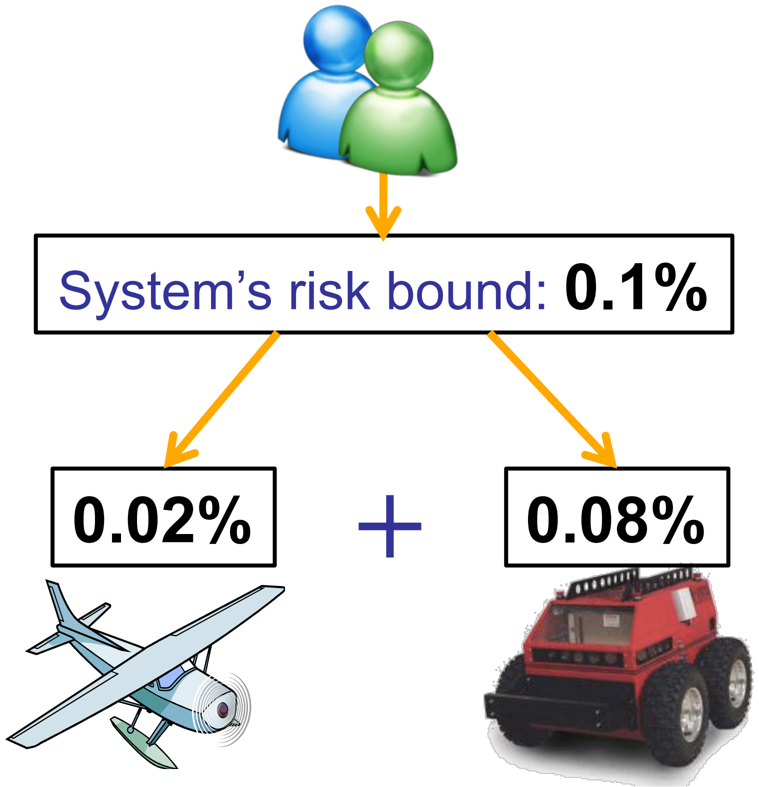
$$\min_{U^{1:I}\in\mathbf{U}^{1:I},\delta_{i:N}^{1:I}} \sum_{i=1}^{I} J^i(U^i)$$

$$s.t.\quad \bigwedge_{i}\bigwedge_{n} h_n^{iT}\overline{X}^i \le g_n^i - m_n^i\left(\delta_n^i\right)$$

$$\sum_{i=1}^{I}\sum_{n=1}^{N^i}\delta_n^i \le \Delta$$

Approach: Decentralized Optimization

User supplies risk by specifying the risk bound



System’s risk bound: **0.1%**

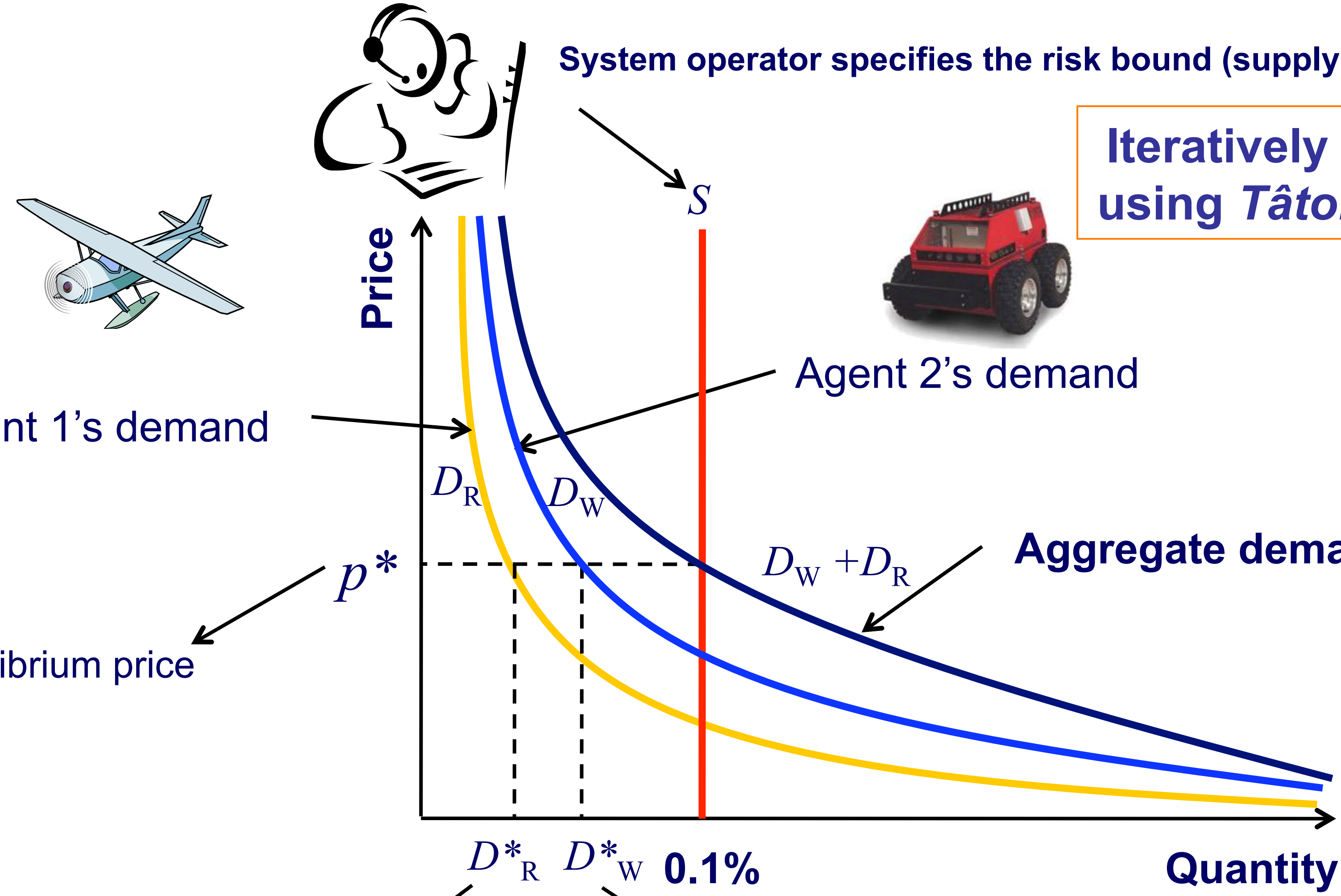
0.02% + **0.08%**

Agents consume risk

$$\sum \left(\textit{Demands for risk} \right) \leq \textit{Supply of risk}$$

- Each agent is an independent decision maker
- Communicates with others
- Finds globally optimal solution through **iterations**
- Inspired by an economic process *tâtonnement*
  - Risk = resource traded in a market
  - Each agent has a *demand for risk* as a function of the price of risk

Market-based Solution to Distributed Risk Allocation (Dual Decomposition)



System operator specifies the risk bound (supply)

Iteratively set price using *Tâtonnement.*

Agent 1's demand

Agent 2's demand

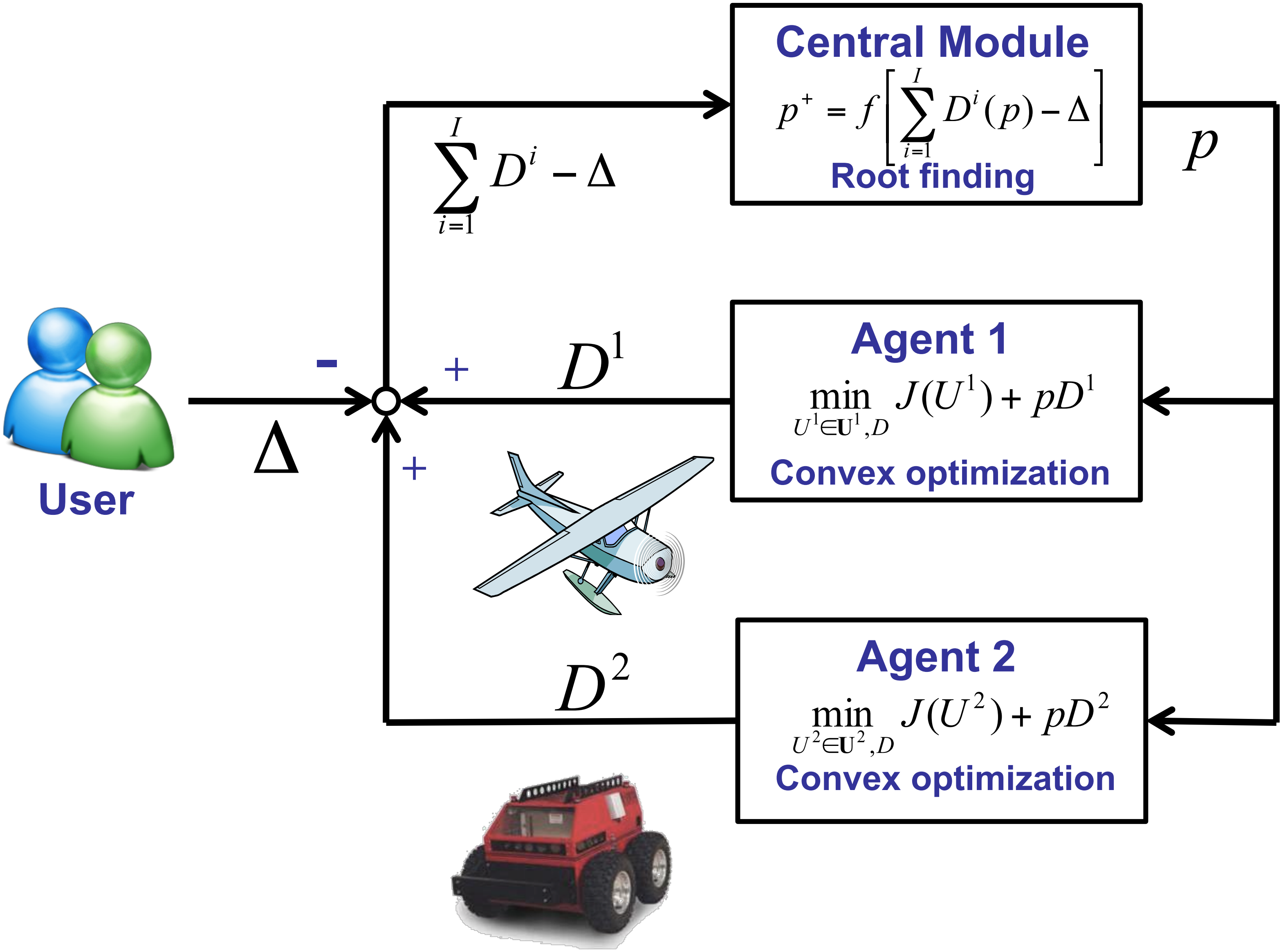
Aggregate demand

Equilibrium price

Risk allocated to Agent 1

Risk allocated to Agent 2

Market-based Iterative Risk Allocation Algorithm



Central Module

$$p^+ = f\left[\sum_{i=1}^{I} D^i(p) - \Delta\right]$$

Root finding

Agent 1

$$\min_{U^1 \in \mathbf{U}^1, D} J(U^1) + pD^1$$

Convex optimization

Agent 2

$$\min_{U^2 \in \mathbf{U}^2, D} J(U^2) + pD^2$$

Convex optimization

Dual Decomposition

Centralized Optimization

(decomposed, deterministic form)

$$\min_{U^{1:I}\in\mathbf{U}^{1:I},\delta^{1:I}_{i:N}} \sum_{i=1}^{I} J^i(U^i)$$

$$s.t. \quad \bigwedge_{i=1}^{I}\bigwedge_{t=1}^{T} x^i_{t+1} = A^i\bar{x}^i_t + B^i u^i_t$$

$$\bigwedge_{i=1}^{I}\bigwedge_{n=1}^{N} h^{iT}_n \bar{X}^i \le g^i_n - m^i_n\left(\delta^i_n\right)$$

$$\sum_{i=1}^{I}\sum_{n=1}^{N^i}\delta^i_n \le \Delta$$

Convex optimization

Solved by Brent's method

De

Risk taken by the *i*'th agent = Demand for risk from *i*'th agent

***i*'th agent: (Primal)**

$$\min_{U^i\in\mathbf{U}^i,\delta^i_{1:N}} J^i(U^i) + \enclose{circle}{p}\boxed{\sum_{n=1}^{N^i}\delta^i_n}$$

Dual variable = Price of risk

$$s.t. \quad \bigwedge_{t=1}^{T} x^i_{t+1} = A^i\bar{x}^i_t + B^i u^i_t$$

$$\bigwedge_{n=1}^{N} h^{iT}_n \bar{X}^i \le g^i_n - m^i_n\left(\delta^i_n\right)$$

Convex optimization

$\delta^{*i}_n(p)$:Optimal solution given p

Central Module (Dual)

$$\sum_{i=1}^{I}\boxed{\sum_{n=1}^{N^i}\delta^{*i}_n(p)} - \Delta = 0$$

Root finding problem

Properties of MIRA

- **Existence of decentralized solution**
  - If the centralized optimization has an optimal solution, it is also an optimal solution for the decentralized optimization
- **Optimality of decentralized solution**
  - If the decentralized optimization has an optimal solution, it is also an optimal solution for the centralized solution
- **Convergence of MIRA**
  - MIRA is guaranteed to converge to an optimal solution if it exists

∴ MIRA is guaranteed to converge to the same solution as the centralized approach

Proofs

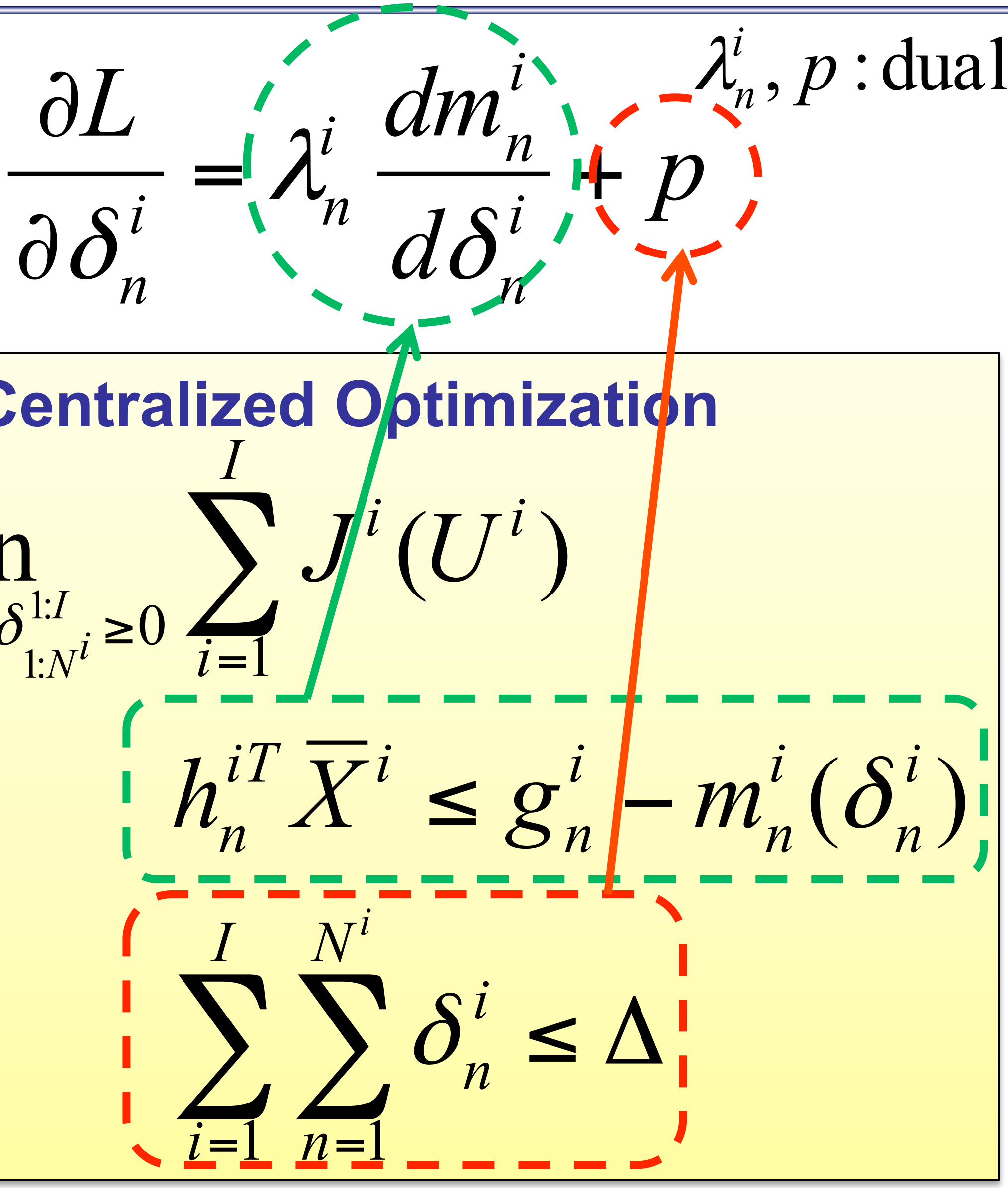
- **Existence**
- **Optimality**

∵ The **KKT conditions** of decentralized optimization coincide with the KKT conditions of centralized optimization

- **Convergence**

∵ Demand functions are **continuous**; Brent's method is guaranteed to converge for continuous functions

Sketch of Proof



KKT stationary condition:

$$\frac{\partial L}{\partial \delta_n^i} = \lambda_n^i \frac{dm_n^i}{d\delta_n^i} + p$$

λ_n^i, p : dual variables

Centralized Optimization

$$\min_{U^{1:I} \in \mathbf{U}^{1:I}, \delta_{1:N^i}^{1:I} \geq 0} \sum_{i=1}^{I} J^i(U^i)$$

$$s.t. \quad h_n^{iT} \overline{X}^i \leq g_n^i - m_n^i(\delta_n^i)$$

$$\sum_{i=1}^{I} \sum_{n=1}^{N^i} \delta_n^i \leq \Delta$$

Sketch of Proof

KKT stationary condition:

$$\frac{\partial L}{\partial \delta_n^i} = \lambda_n^i \frac{dm_n^i}{d\delta_n^i} + \boxed{p}$$

MIRA (each agent)

$$\min_{U^i \in \mathbf{U}^i, \delta^i_{1:N^i} \geq 0} \quad J^i(U^i) + \boxed{p \sum_{n=1}^{N^i} \delta_n^i}$$

$$s.t. \qquad h_n^{iT} \overline{X}^i \leq g_n^i - m_n^i(\delta_n^i)$$

$$\sum_{i=1}^{I} \sum_{n=1}^{N^i} \delta_n^i \leq \Delta$$

Sketch of Proof

KKT comp. slackness condition:

$$p\left(\sum_{i=1}^{I}\sum_{n=1}^{N^i}\delta_n^i - \Delta\right) = 0$$

Centralized Optimization

$$\min_{U^{1:I}\in\mathbf{U}^{1:I},\,\delta_{1:N^i}^{1:I}\geq 0} \sum_{i=1}^{I} J^i(U^i)$$

$s.t.$

$$h_n^{iT}\overline{X}^i \leq g_n^i - m_n^i(\delta_n^i)$$

$$\sum_{i=1}^{I}\sum_{n=1}^{N^i}\delta_n^i \leq \Delta$$

Sketch of Proof

KKT comp. slackness condition:

$$p\left(\sum_{i=1}^{I}\sum_{n=1}^{N^i}\delta_n^i - \Delta\right) = 0$$

MIRA (central module)

$$\sum_{i=1}^{I}\sum_{n=1}^{N^i}\delta_n^{*i}(p) - \Delta = 0$$

- Special case with $p=0$ is handled separately

Proofs

- **Existence**
- **Optimality**

∵ The **KKT conditions** of decentralized optimization coincide with the KKT conditions of centralized optimization

- **Convergence** ∵ Demand functions are **continuous**; Brent's method is guaranteed to converge for continuous functions

Definition: Cost of Risk for i'th Agent

$$
\begin{aligned}
J^{i\star}(\Delta^i) := \min_{\boldsymbol{U}^i, \delta^i_{1:N^i} \geq 0} \quad & J^i(\boldsymbol{U}^i) \\
\text{s.t.} \quad & \bar{\boldsymbol{x}}^i_{t+1} = \boldsymbol{A}^i \bar{\boldsymbol{x}}^i_t + \boldsymbol{B}^i \boldsymbol{u}^i_t \\
& \boldsymbol{u}^i_{\min} \leq \boldsymbol{u}^i_t \leq \boldsymbol{u}^i_{\max} \\
& h^{iT}_n \bar{\boldsymbol{X}}^i \leq g^i_n - m^i_n(\delta^i_n) \\
& \sum_{n=1}^{N^i} \delta^i_n \leq \Delta^i
\end{aligned}
$$

$J^{i\star}(\Delta^i)$: minimum cost the agent can achieve when it is allowed to take up to Δ^i of risk in total

Each Agent’s Optimization Problem

$$\begin{array}{ll} \displaystyle\min_{\boldsymbol{U}^i, \delta^i_{1:N^i} \geq 0} & J^i(\boldsymbol{U}^i) + p \sum_{n=1}^{N^i} \delta^i_n \\ \text{s.t.} & \bar{\boldsymbol{x}}^i_{t+1} = \boldsymbol{A}^i \bar{\boldsymbol{x}}^i_t + \boldsymbol{B}^i \boldsymbol{u}^i_t \\ & \boldsymbol{u}^i_{\min} \leq \boldsymbol{u}^i_t \leq \boldsymbol{u}^i_{\max} \\ & h^{iT}_n \bar{\boldsymbol{X}}^i \leq g^i_n - m^i_n(\delta^i_n) \\ & (t = 0 \cdots T-1, n = 1 \cdots N^i) \end{array}$$

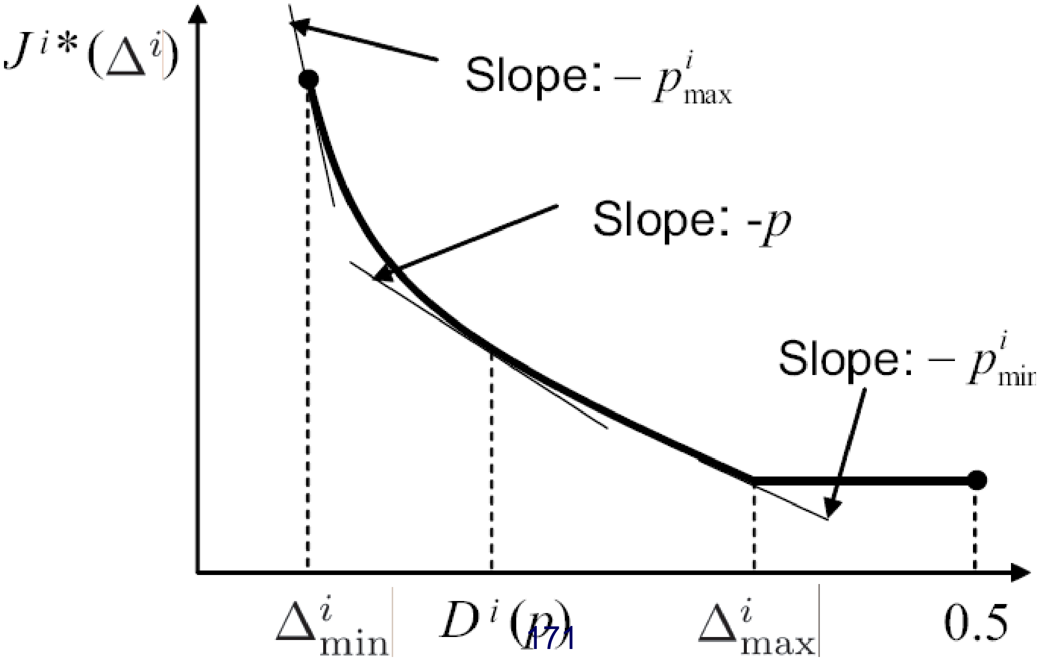
$$\|$$

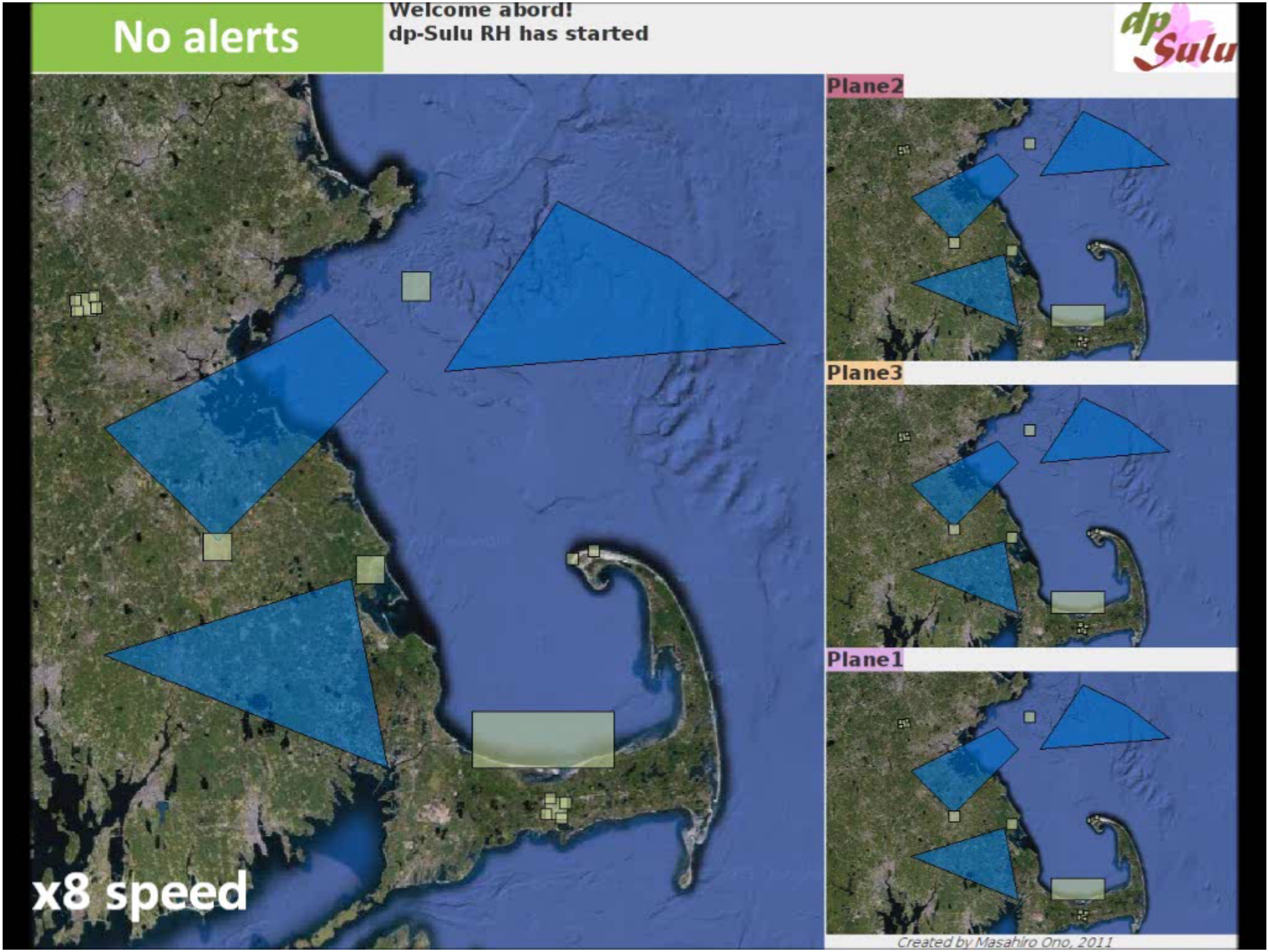
$$\min_{\Delta^i \leq 0.5} J^{i\star}(\Delta^i) + p\Delta^i$$

Sketch of Proof

Starting from: convexity of $J^i(\boldsymbol{U}^i)$ (assumption)

1. $J^{i\star}(\Delta^i)$ is monotonically decreasing, *strictly* convex
   - strict convexity of $m_n^i(\delta_n^i)$ (inverse of cdf of Gaussian)
2. $D^i(p)$ is continuous
   - Conjugate Subgradient Theorem (Bertsekas 2009)





No alerts

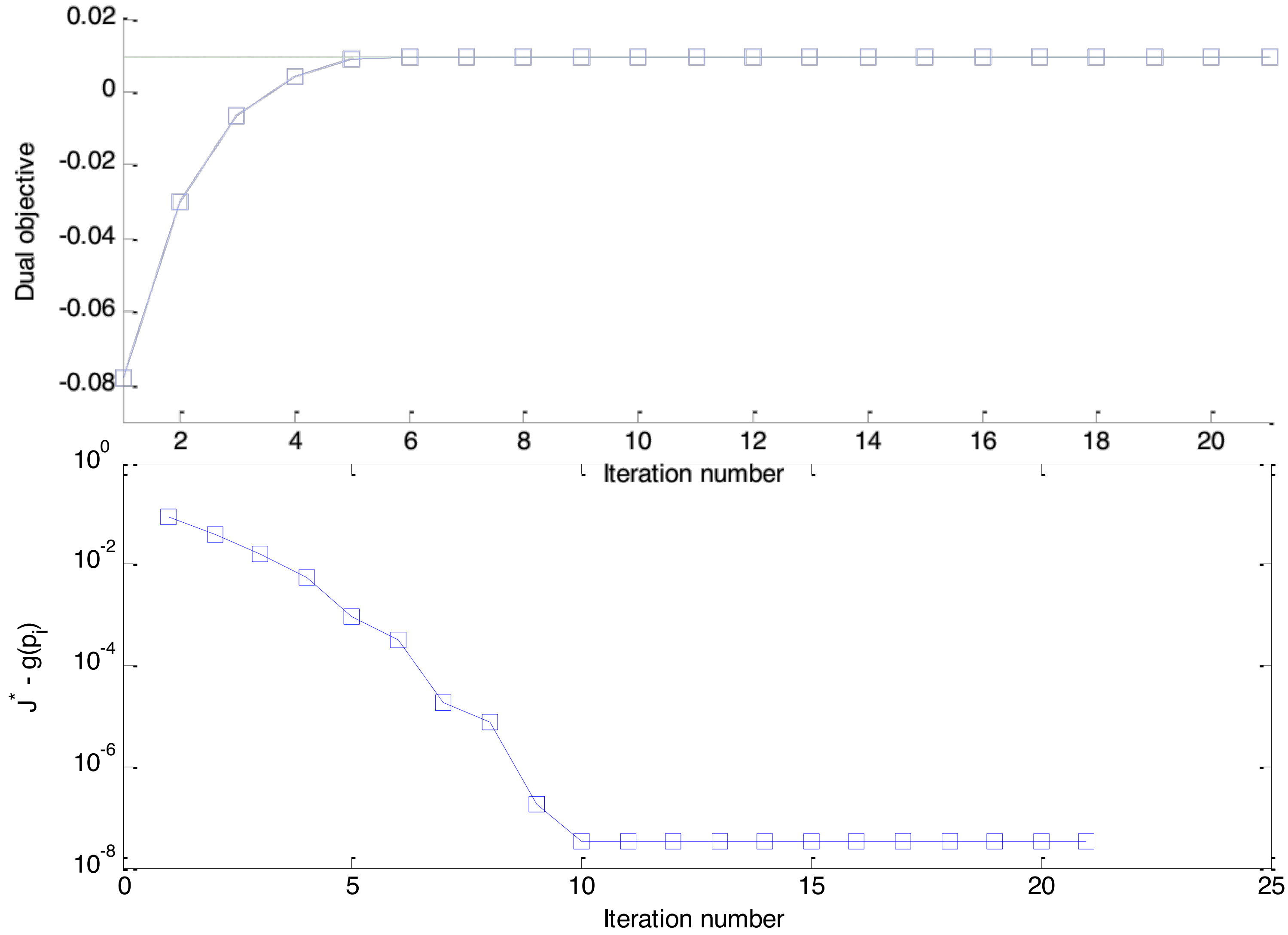
Welcome abord!
dp-Sulu RH has started

Created by Masahiro Ono, 2011

Convergence to Optimal Solution

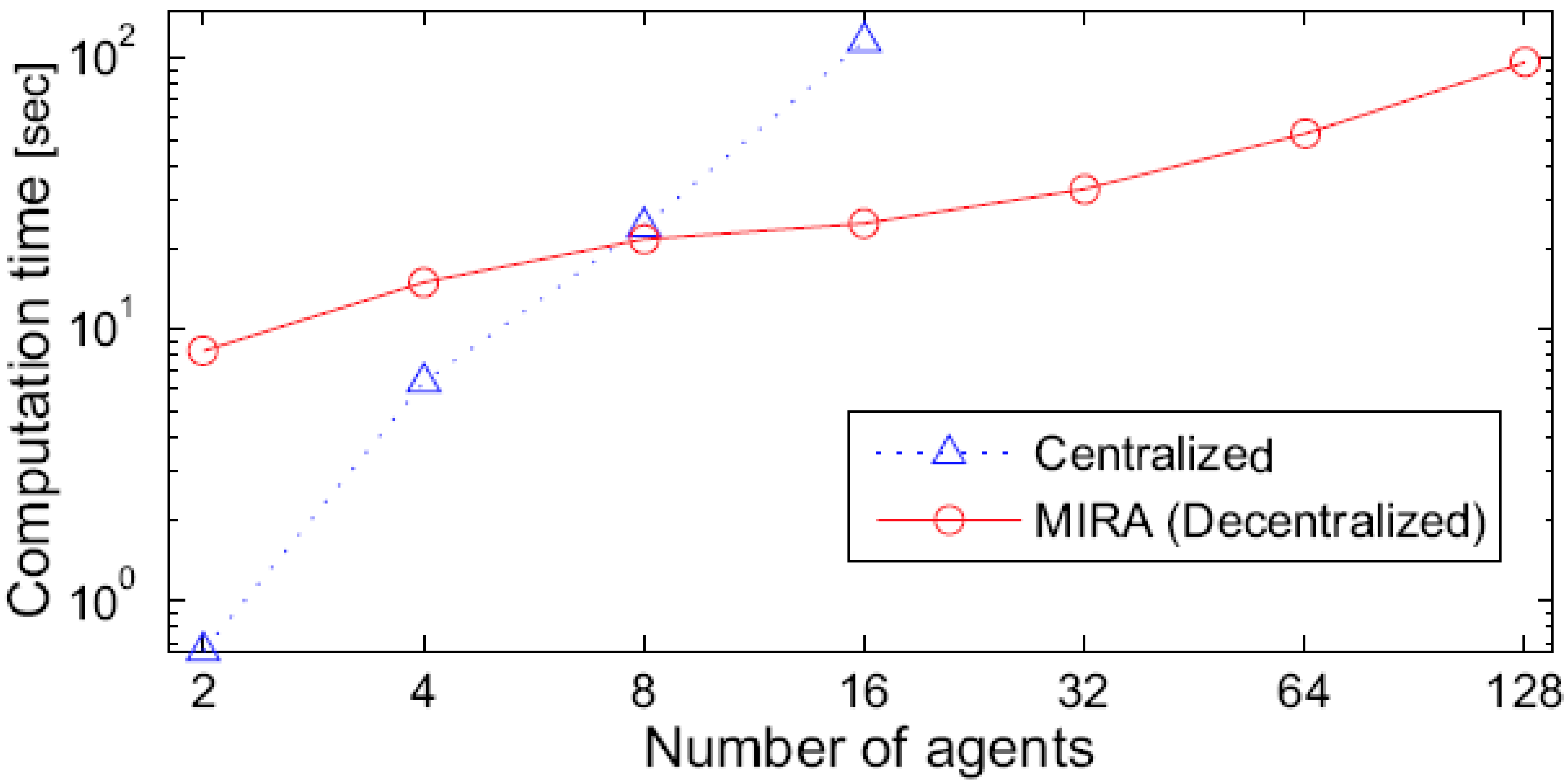
Dual objective value

Duality gap



16 agents, $\Delta=0.01$, T=5

Result: Scalability



Values are the averages of 100 runs each

$$A^i = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, B^i = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}, x_0^i = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}, \quad \Sigma_w = \begin{bmatrix} 0.001 & 0 \\ 0 & 0 \end{bmatrix}.$$

$$u_{\min} = -0.2, u_{\max} = 0.2, \quad h_n^i = -[1 \quad 0] \qquad J^i = \sum_{t=1}^{5} (u_t^i)^2$$