Motion Planning for Dynamic Environments

Part III - Dynamic Environments: Modeling Issues

Steven M. LaValle
University of Illinois

Solution to Homework 2

For an infinite sample sequence $\alpha : \mathbb{N} \to X$, let α_k denote the first k samples.

Find a metric space $X \subseteq \mathbb{R}^n$ and α so that:

1. The dispersion of α_k is ∞ for all k.
2. The dispersion of α is 0.

Solution:

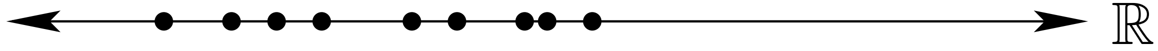
Solution to Homework 2

For an infinite sample sequence $\alpha : \mathbb{N} \rightarrow X$, let α_k denote the first k samples.

Find a metric space $X \subseteq \mathbb{R}^n$ and α so that:

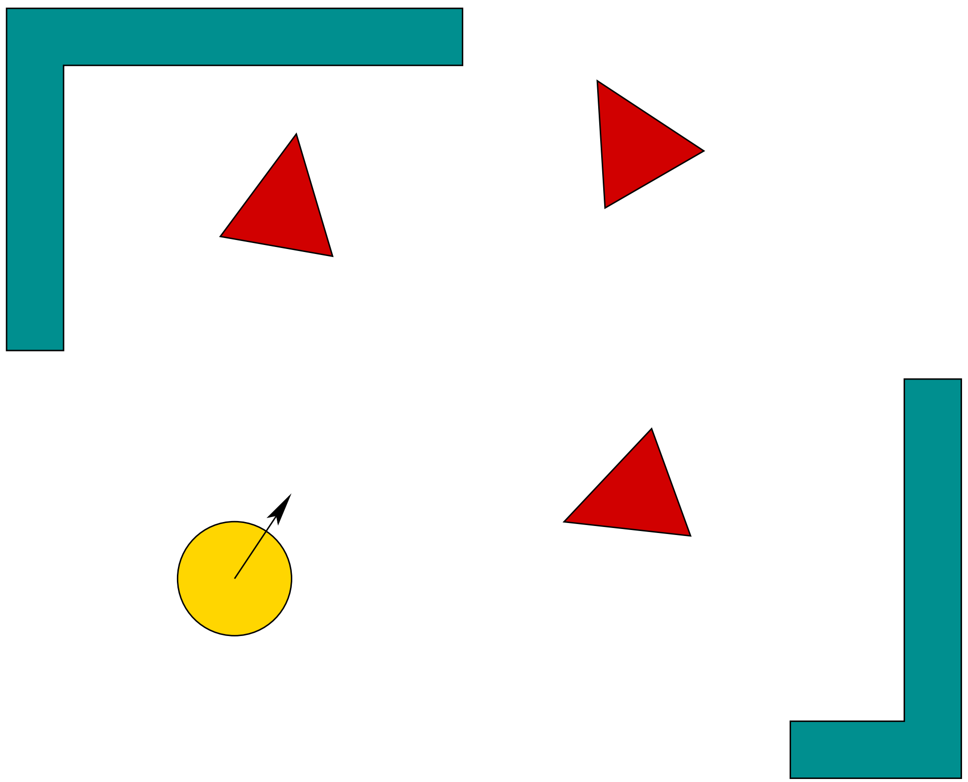
1. The dispersion of α_k is ∞ for all k.
2. The dispersion of α is 0.

Solution: $X = \mathbb{R}$ and α is any dense sequence.
Example: An enumeration of $\mathbb{Q}$.
For any finite sample set in $\mathbb{R}$, the dispersion in finite.



Basic Choices

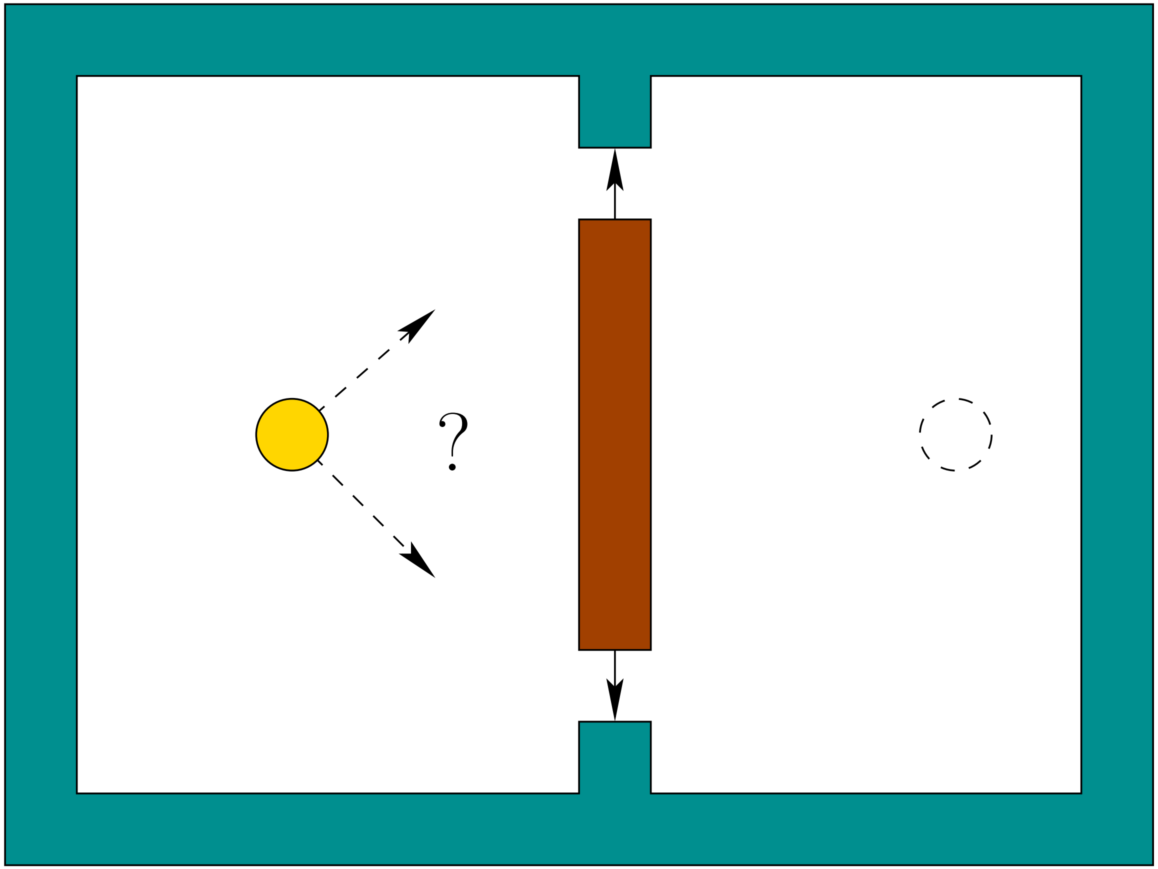
A robot moves among static and moving obstacles:



How to model this?

The time axis $T = [0, t_f]$ becomes critical, unlike before.

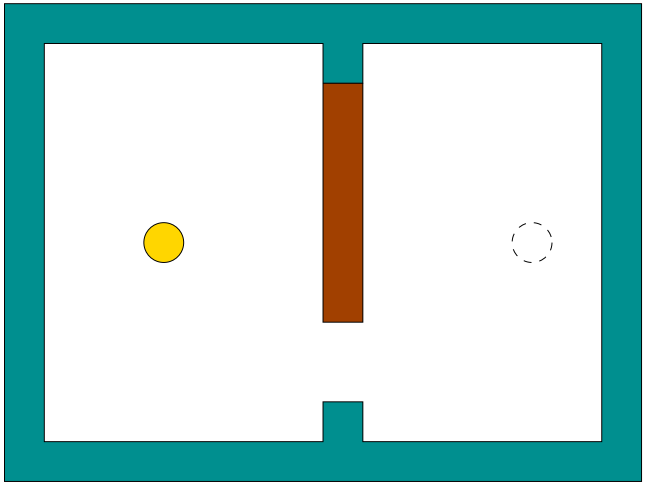
A thought experiment



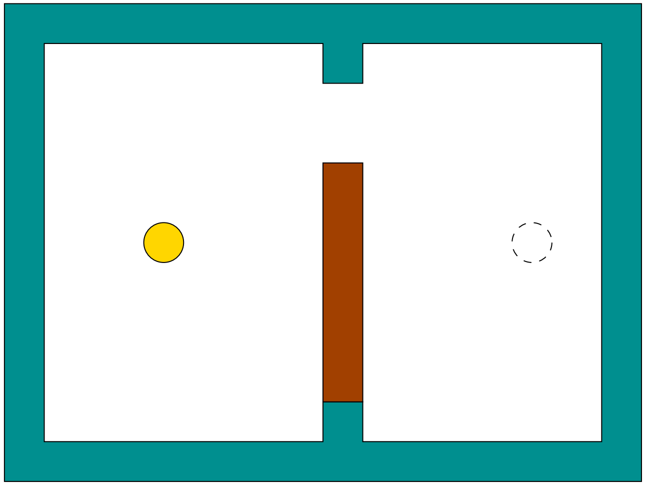
There is an autonomous sliding door.

Where should the robot move?

Fundamental Limitations



Mode 1



Mode 2

- There is no solution if an adversarial “demon” moves the wall.
- Iterative replanning leads to oscillation.
- If wall position follows a Markov chain, then wait by either potential opening.

Match the Model to the World

What might this correspond to in the world?

In a complex environment, what is the concern?

- Stationary obstacles
- Hazardous terrain
- Moving people, animals, vehicles, other robots

Is the environment friendly, hostile, oblivious?

A *state space* X should capture all contingencies.

Components of a state $x \in X$:

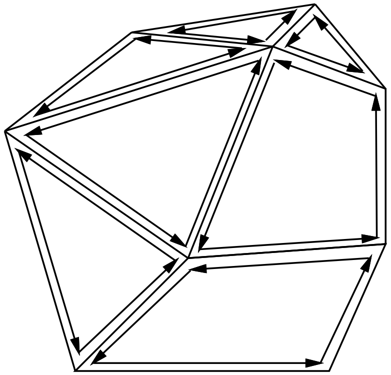
- The robot C-space $\mathcal{C}$ or phase space.
- The C-space or phase space of other controllable robots?
- The C-space or phase space of moving obstacles?
- The C-space or phase space of other agents?
- Possible modes for the environment?

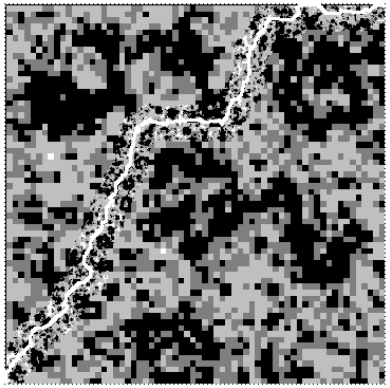
Can a space of states be nicely parametrized?
What components are *predictable*?
What components are *sensable*?

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Representation Issues

The environment modeling issues return





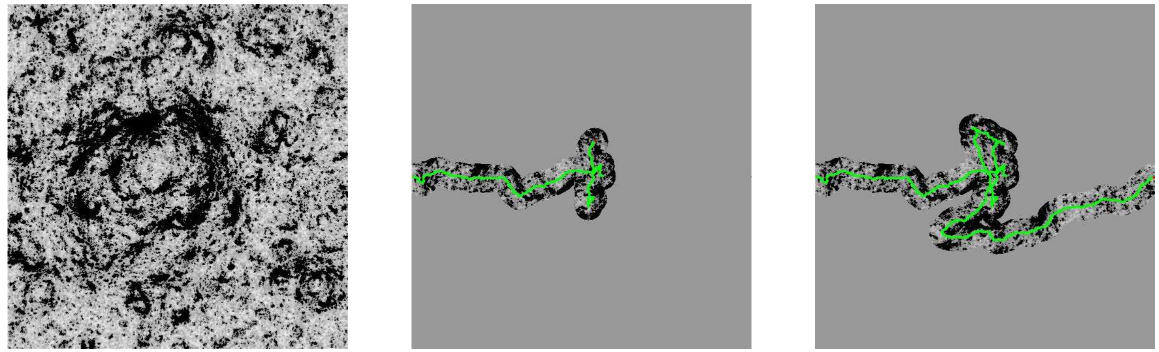
But now time-critical issues may dominate.

Important to obtain timely updates to the map.

As time becomes critical, what representation is minimally needed?

Partial vs. Complete Map

What happens if there are unknown regions in the map?



(figures from Tony Stentz)

The environment might seem “dynamic” because a static obstacle is revealed.

What assumptions are made for the invisible part of the environment?

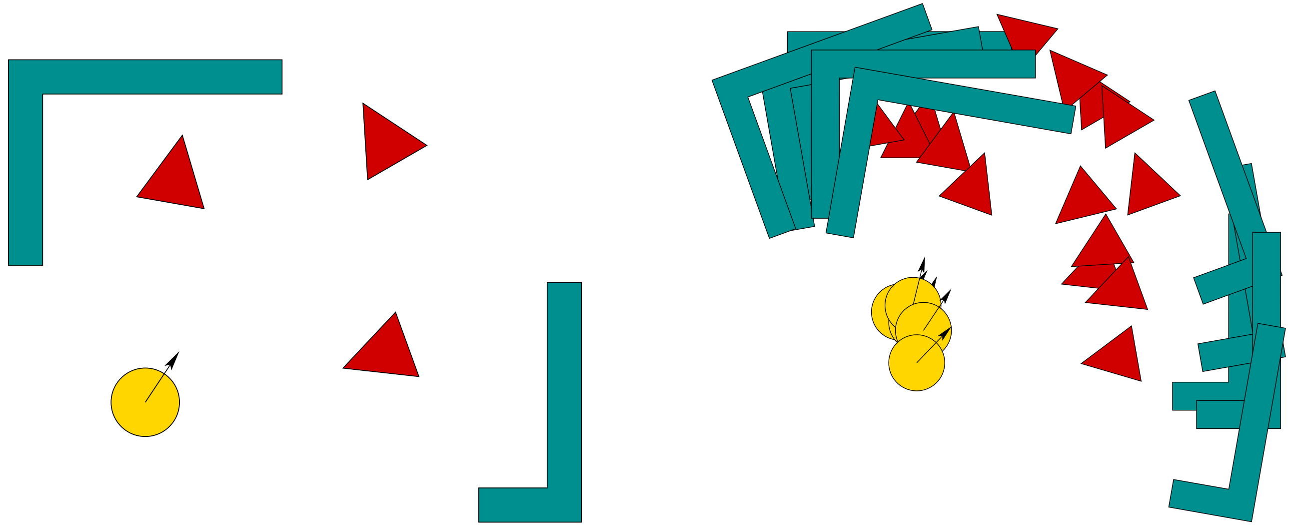
Optimism, pessimism, or maximum likelihood?

Dynamic Environments

What is the *environment*?

Suppose the robot itself is controlled by a stochastic law: $p(q_{k+1}|q_k, u_k)$

The obstacle locations in the robot's frame depend on disturbance in the robot motion.



Present state   Possible futures

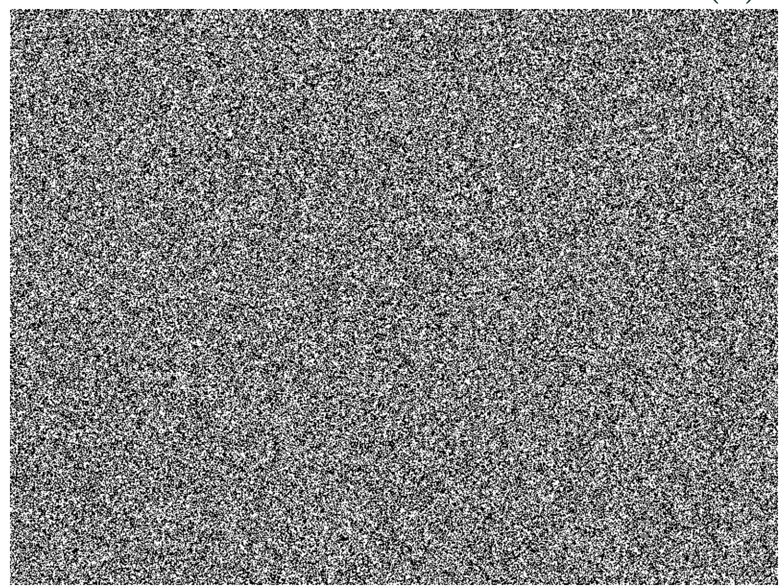
Only the robot-centric frame moves, so the outside motions are rigidly coupled through $SE(2)$ or $SE(3)$.

Think robotic relativity!

Completely Unpredictable Environments

At one extreme, the environment may be **completely unpredictable**.

Example: Suppose that at any time t any subset $\mathcal{O}(t) \subset \mathbb{R}^3$ is possible.



Is this realistic? Not really. A robot could only panic.

In reality, we at least have **partial predictability**.

Various Levels of Predictability

Consider these cases for a rigid obstacle:

1. It moves along a known trajectory.
2. It moves in any direction with bounded speed.
3. It follows $\dot{q} = f(q, \theta)$ for some unknown $\theta \in \Theta$.
4. It follows $p(q_{k+1}|q_k)$, a stochastic transition model.
5. It follows $p(q_{k+1}|q_k, \theta_k)$ for some unknown $\theta_k \in \Theta$.
6. It appears according to a Poisson process.
7. An intention-based model.
8. Game-theoretic models

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Fully Predictable

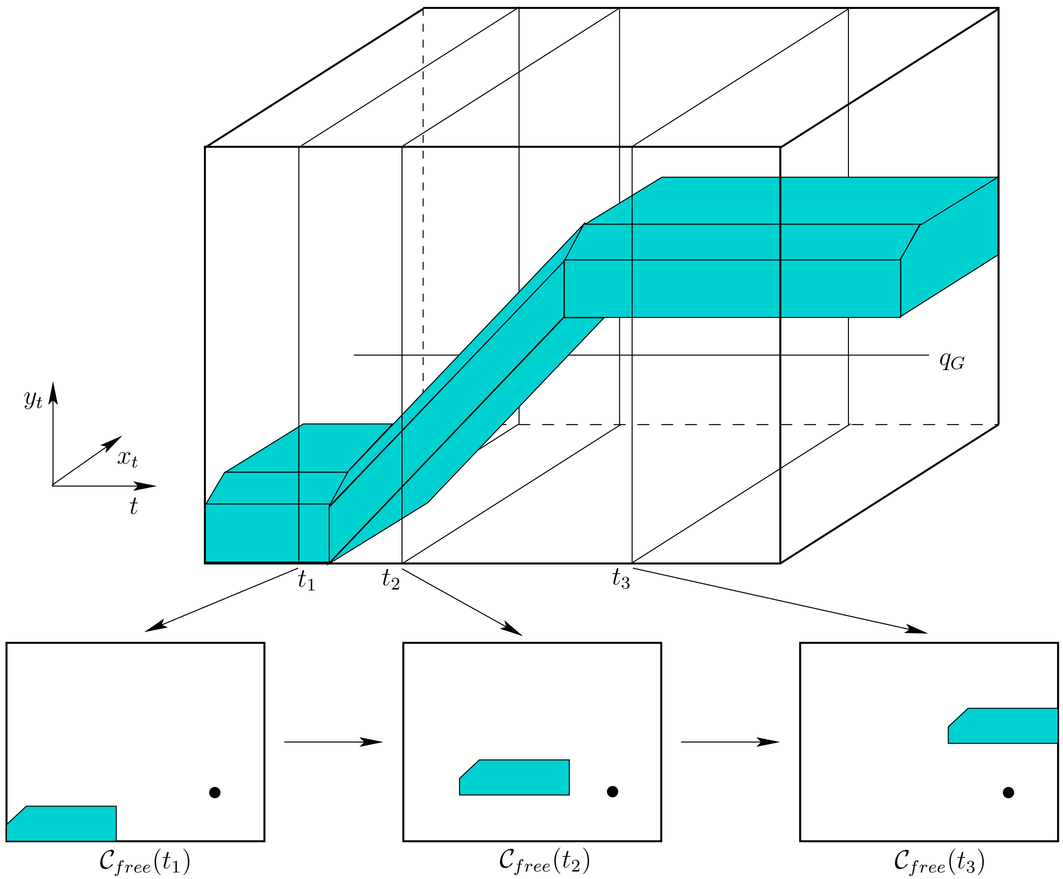
Predictable Obstacles

Let $T = [0, t_f]$ denote a time interval of interest.

Let $\mathcal{O}(t) \subset \mathcal{W}$ denote the obstacle at time $t \in T$.
Assume $\mathcal{O}(t)$ is given for all $t \in T$.

Let $Z = \mathcal{C} \times T$ denote the *configuration-time* space.
Each $(q, t) \in Z$ specifies both $\mathcal{A}(q)$ and $\mathcal{O}(t)$.

Obstacles in Configuration-Time Space



At each time slice $t \in T$, we must avoid

$$\mathcal{C}_{obs}(t) = \{q \in \mathcal{C} \mid \mathcal{A}(q) \cap \mathcal{O}(t) \neq \emptyset\}$$

Finding a Collision-Free Path

Let

$$Z_{obs} = \{(q,t) \in X \mid \mathcal{A}(q) \cap \mathcal{O}(t) \neq \emptyset\},$$

and $Z_{free} = Z \setminus Z_{obs}$.

Initial state: $z_{init} = (q_I, 0)$.
Goal region $Z_{goal} \subset Z_{free}$ (a combination of time and configuration).

Problem: Compute a continuous *trajectory*

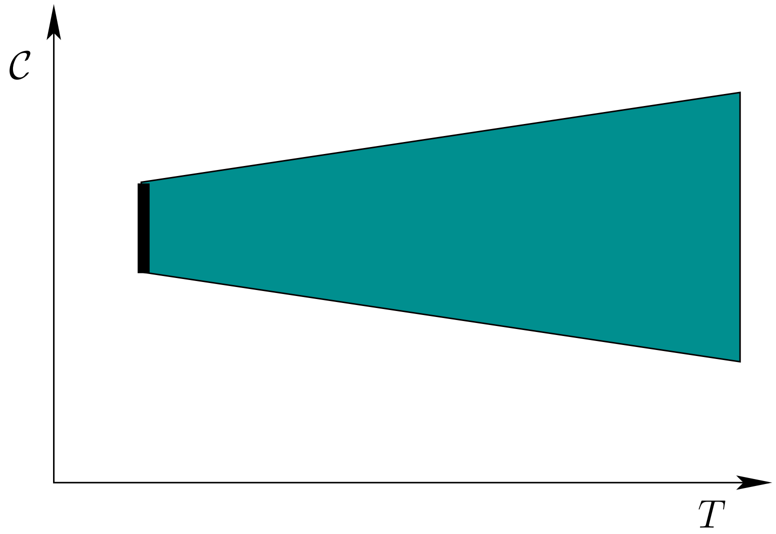
$$\tau : T \rightarrow Z_{free}$$

so that $\tau(0) = z_{init}$ and $\tau(t) \in Z_{goal}$ for some $t \in T$.
Note: A trajectory is a time-parametrized path.

More challenging case: The robot has a maximum speed bound
Even more challenging: Robot motion is specified as a nonlinear system

Bounded Uncertainty

Let one moving obstacle be called a *body*.
The body moves with a maximum speed bound: $\|v\| \leq c$.



Using bounded uncertainty models, we once again reason in configuration-time space Z.
This is called a *reachable set* computation.

Determine a safe $q \in \mathcal{C}_{free}(t)$ for every future t.
Find a trajectory $\tau : T \rightarrow Z_{free}$.

Bounded Uncertainty Models

Could overapproximate $\mathcal{C}_{obs}(t)$: Conservative bounds fine, but lose completeness.

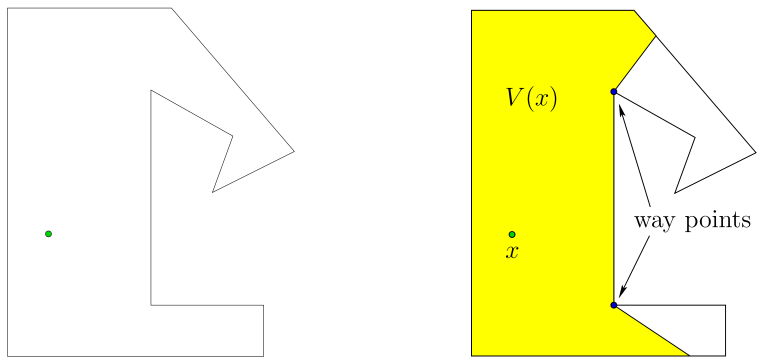
What should happen if sensors can tell current obstacle locations during execution?

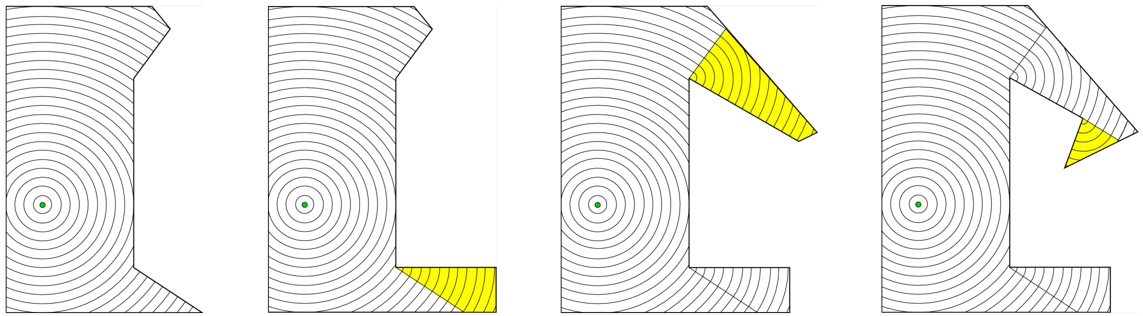
- If there was a solution from the initial time, then on-line information is not necessary.
- The problem may initially appear unsolvable, but on-line information could make it solvable.
- It is tempting to try a replanning approach.

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Suppose that the moving body is a point.

Its possible future positions can be calculated by the *continuous Dijkstra* algorithm (Hershberger, Suri, 1995; Mitchell 1996)





What if the the body is a polygon?
If it translates with bounded speed, then continuous Dijkstra could be used in C-space.
But not too realistic.

More generally, suppose the body moves according to a control law:

$$\dot{x}_b = f(x_b, u_b),$$

in which x_b is its *state* and $u_b \in U_b$ is the *input* being applied to it.

Reachable Set Computations

The obstacle has its own configuration space $\mathcal{C}_b$.

Each state x_b in the state space X_b represents

$$x_b = (q_b, \dot{q}_b)$$

in which $q_b \in \mathcal{C}_b$ is the configuration and $\dot{q}_b$ is the velocity.

Each input u_b could represent an unknown control signal or disturbance.

From a given $\tilde{u}_b : [0, t] \rightarrow U_b$ and initial state $x_b(0)$, the state at time t is given by

$$x_b(t) = x_b(0) + \int_0^t f(x_b(t'), u_b(t'))dt'.$$

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Reachable Set Computations

Problem: Determine *all possible* $x_b(t)$ if $\tilde{u}_b$ is not given!

Let $R(x, t)$ denote the *time-limited reachable set* from initial state x and at time t.

A state $x_b \in R(x, t)$ if and only if there exists a control $\tilde{u}_b$ which upon integration from state x causes the system to arrive at x_b at time t.

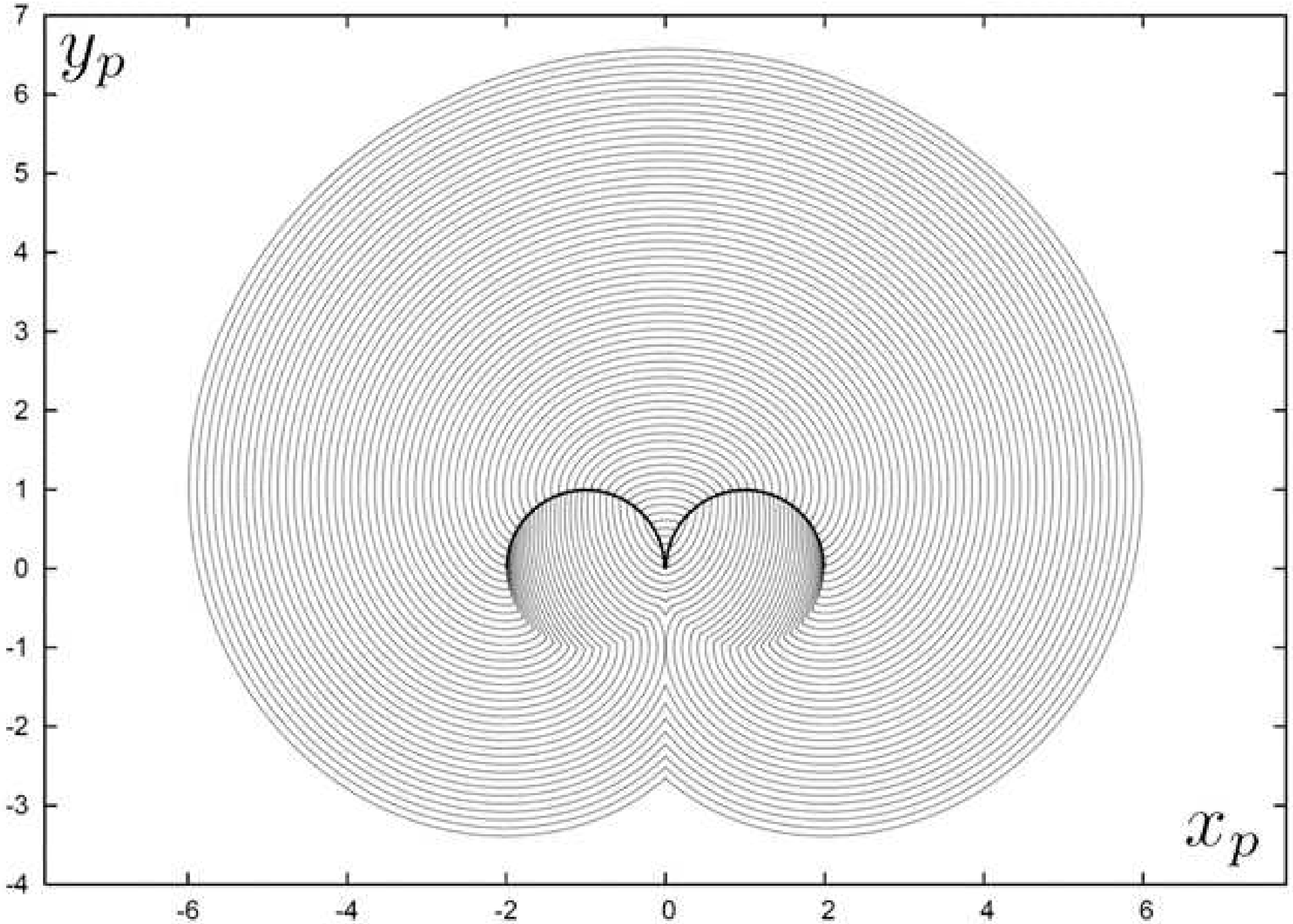
From $R(x, t)$ we could obtain the set of possible $\mathcal{O}(t)$ due to the body. From this we need to compute $\mathcal{C}_{obs}(t)$ for the robot's perspective.

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Reachable Set Computation Methods

Reachable sets for the Dubins car:

$$\dot{x}_p = \sin\theta$$
$$\dot{y}_p = \cos\theta$$
$$\dot{\theta} = u$$
$$|u| \leqslant 1$$



From Patsko, Turova, 2008.

Reachable Set Computation Methods

It is a key problem in system *verification*.
Make sure the right behavior is obtained for *all* inputs.

- For linear systems: Incremental Minkowski sums and zonotopes
- For nonlinear systems: Hamilton-Jacobi PDE computations
- Also possible: Monte Carlo simulation

See works of Oded Maler (Verimag), Claire Tomlin (Berkeley), Ian Mitchel (UBC),

Note: Verification is “anti-planning”:
Try to prove that there does not exist a path.

Alternative, from the robot's perspective.

Robot control system:

$$\dot{x} = f(x, u)$$

in which $x = (q, \dot{q})$ and $u \in U$ is the input set.

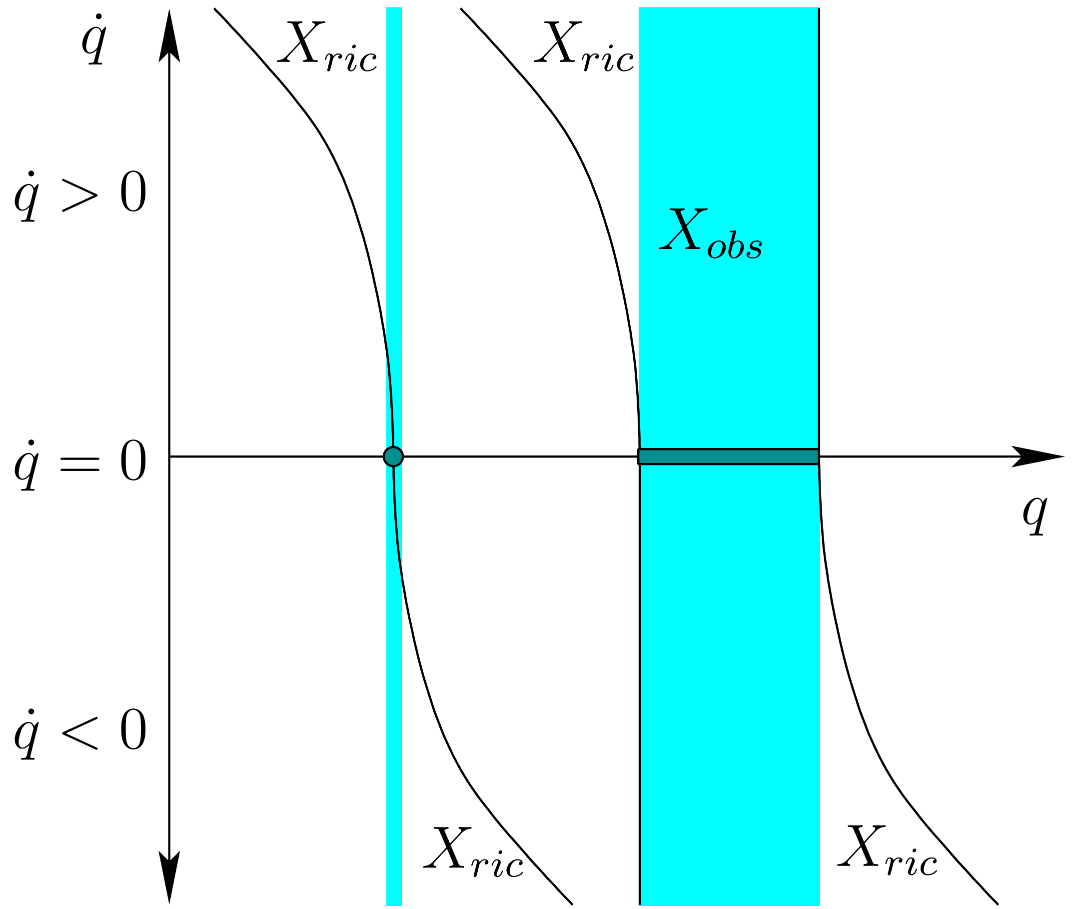
Suppose the obstacles are static (not necessary).

Let X_{obs} denote the set of states $x = (q, \dot{q})$ for which $q \in \mathcal{C}_{obs}$.

The *region of inevitable collision* is

$$X_{ric} = \{x(0) \in X \mid \text{ for any } \tilde{u}\ ,\ \exists t > 0 \text{ such that } x(t) \in X_{obs}\}.$$

Region of Inevitable Collision



Computation methods for safe planning: Fraichard, Asama, 2004; Bekris, Kavraki, 2008; Chan, Kuffner, Zucker, 2008.

Probabilistic Uncertainty

Probabilistic Models

Rather than bounded uncertainty, suppose that a density

$$p(x_b' \mid x_b)$$

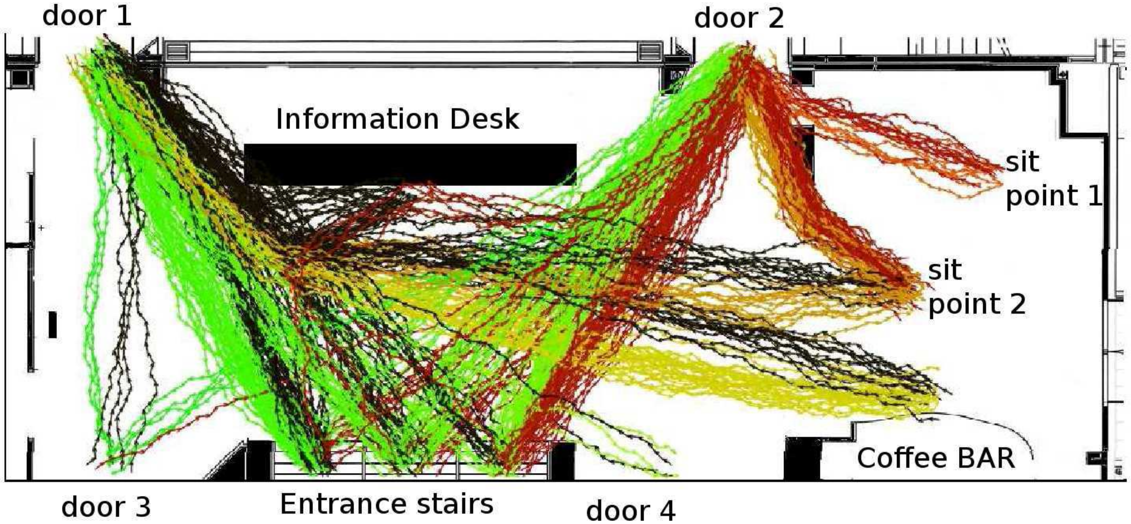
is known.

x_b is the body state at time t
x_b' is the body state at time $t + \Delta t$

Where might the body go next?

- Simple diffusion models
- Brownian motions
- Could calculate with particle filters

Perhaps a model can be learned from data.

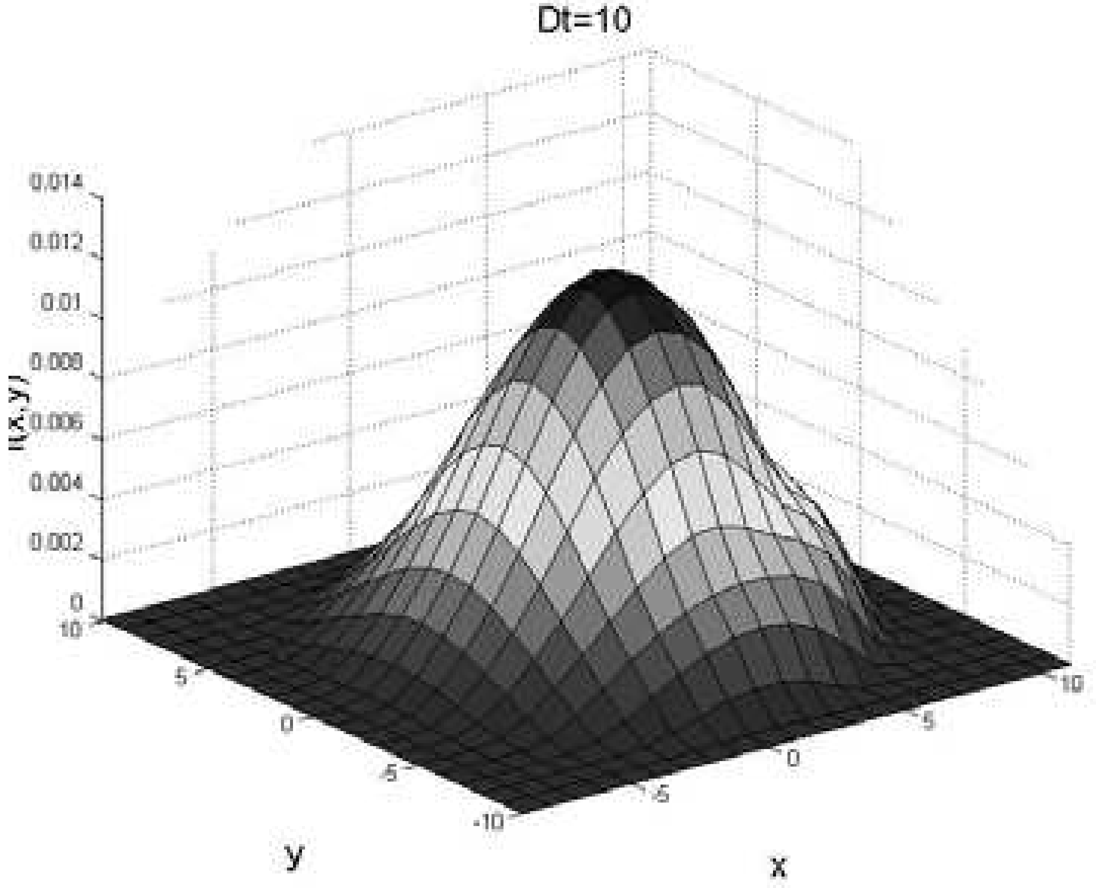


(from Chiara Fulgenzi, INRIA)

Intentions become important to reduce model complexity.
Could learn a Hidden Markov Model (HMM) that captures positions, velocities, and intentions of obstacles.
Could develop sampling-based (particle) representations of future obstacle trajectories.

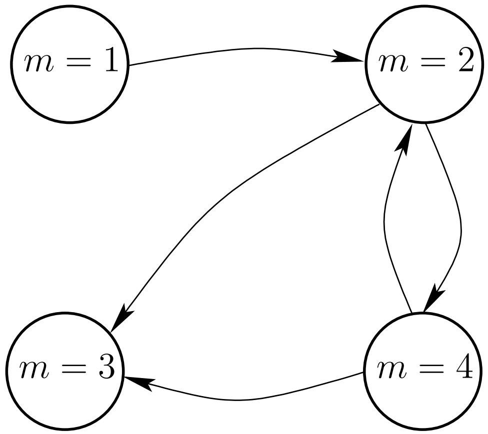
Perhaps the body is another robot and its intentions are known.

Might nevertheless have to use stochastic models to predict the future behavior.

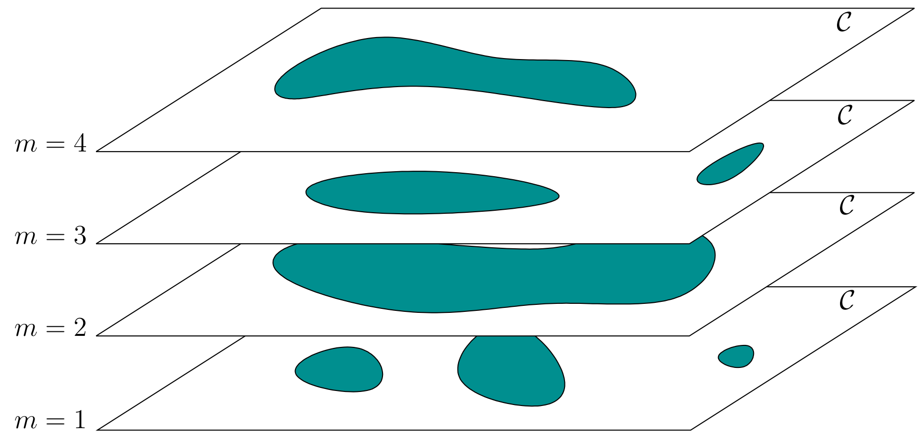


(from Zhou, Chirikjian, 2003)

Could have continuous layers and switching between discrete modes.



Modes

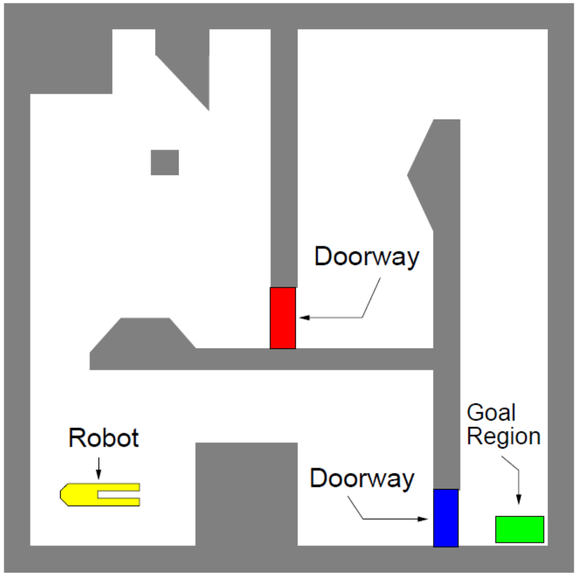


Layers

In each layer, the obstacle region is different.

A Markov chain models the mode transitions: $P(m'|m)$

Hybrid System Model



Examples of modes:

- Doors may open or close
- An obstacle may appear or disappear
- A hazardous situation may emerge: rain falling, giant vehicle approaching
- Sudden task change

Game Theory

So far, the obstacle or body motions did not depend on the robot.

Generally have two or more "players" with one transition equation.
Here's suppose there are two players (the "robot" and "obstacle")

In continuous time, the state x includes both player configurations (and possibly velocities).

Player 1 chooses actions $u \in U$ and Player 2 chooses $v \in V$.

In continuous time, a *differential game* is obtained:

$$\dot{x} = f(x, u, v).$$

In discrete time, a *sequential game* is obtained:

$$x_{k+1} = f(x_k, u_k, v_k).$$

If Player 1 is the robot, then it must choose u_k in each iteration to ensure that no collision occurs.

Also, could optimize objectives, such as reaching the goal in few steps.

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

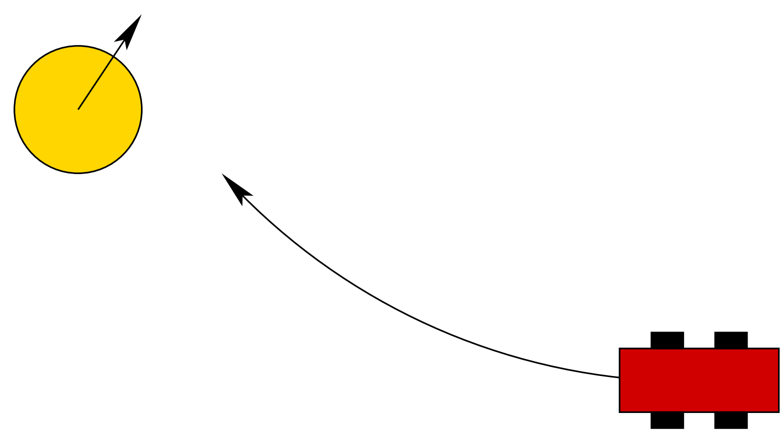
Example: Homicidal Chauffeur

A car-like robot vs. omnidirectional human (Isaacs, 1951)

$$\dot{x}_1 = s_1 \cos\theta_1 \qquad \dot{x}_2 = s_2 \cos v$$

$$\dot{y}_1 = s_1 \sin\theta_1 \qquad \dot{y}_2 = s_2 \sin v$$

$$\dot{\theta}_1 = \frac{s_1}{L} \tan u_\phi$$



See also: Homicidal diff. drive robot, Ruiz, Murrieta-Cid, ICRA 2012

Game Theory Troubles

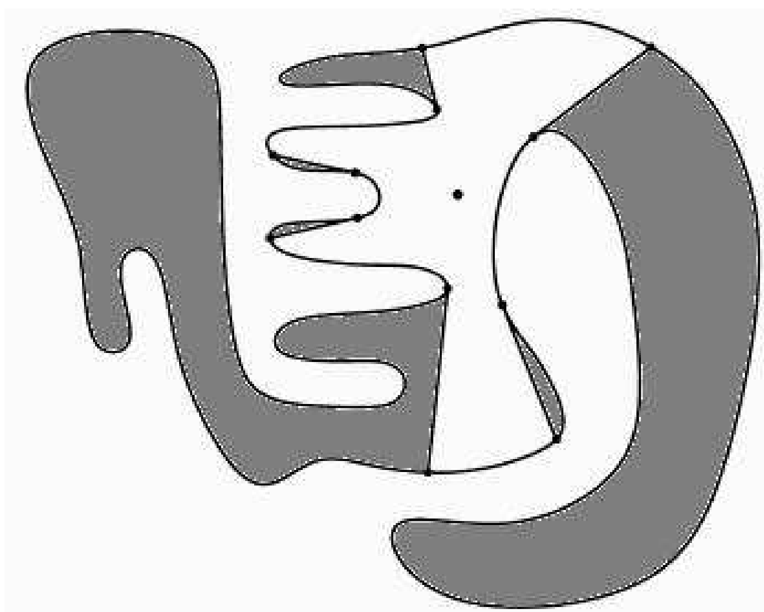
Are the models correct for other players?

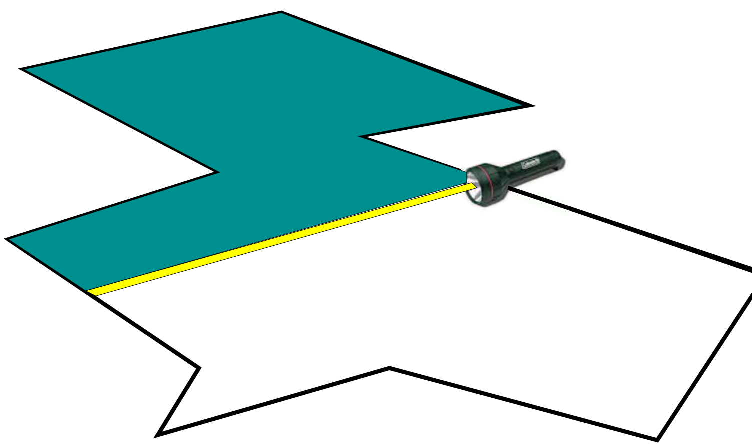
For planning in dynamic environments, we need to know:
Under what conditions can collision be avoided?

Unfortunately, this returns to reachable set computations.
Perfect characterizations exist only for simple cases.
Additional obstacles complicate everything.

In some cases, a simpler model may help dramatically.

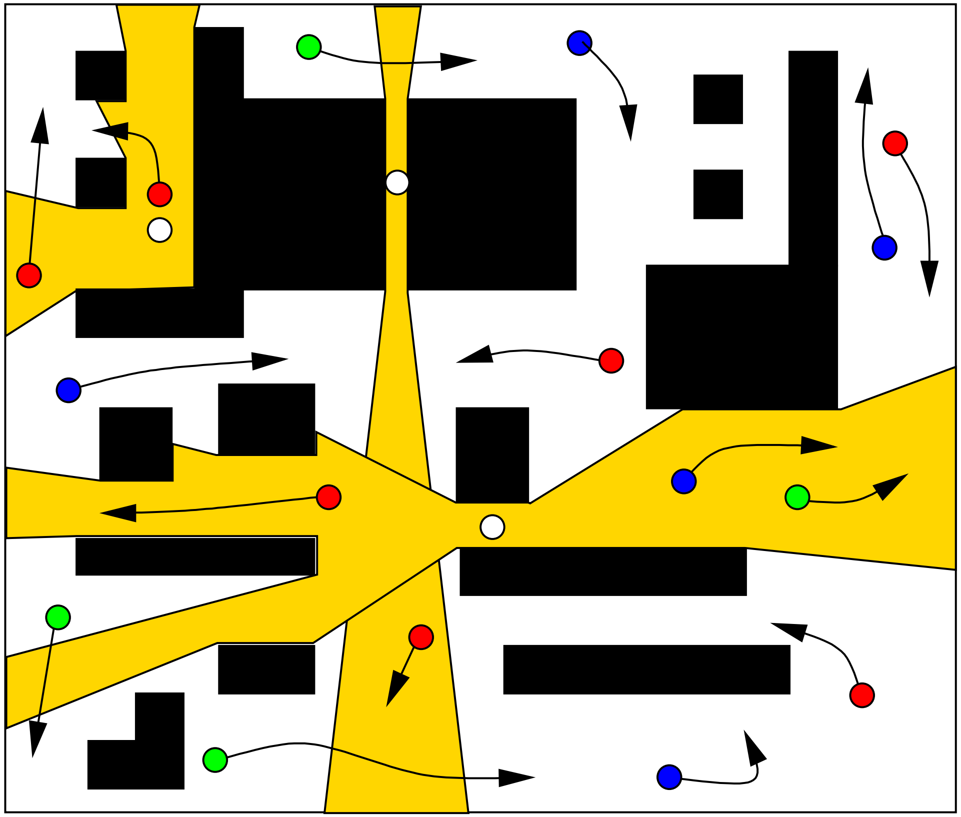
- A 2D environment, possibly curved
- Unpredictable point “evaders” move with unbounded speed
- Point “pursuers” use visibility sensors to find all evaders





Introduced by Suzuki, Yamashita, 1992.

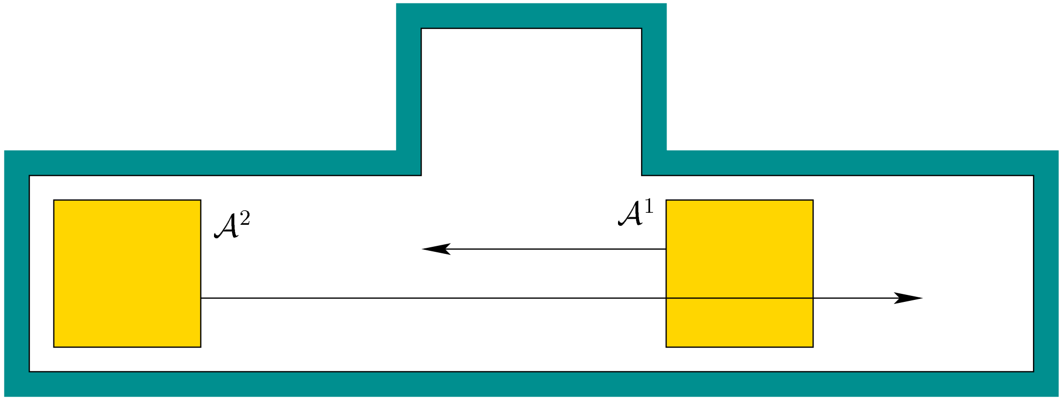
More General: Shadow Information Spaces



Keep track of bodies out of view–in the shadows.

How many are there? What kinds are there?

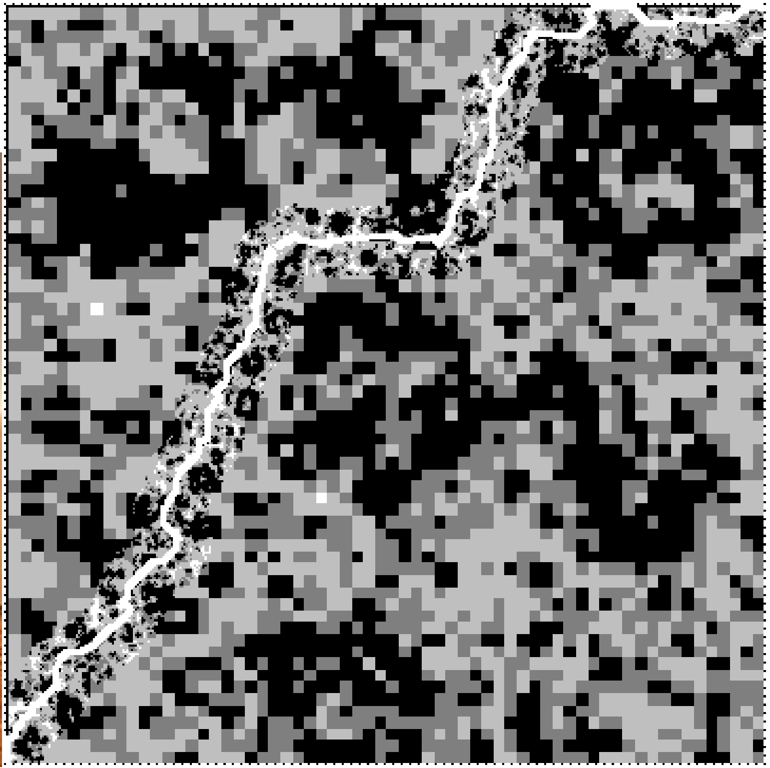
- Law of the jungle: Smaller bodies move out of the way. (what if they become trapped?)
- Multiple robot coordination, but limited communication. (protocols may become important)
- Fully cooperative setting: Pareto optimal coordination.



Non-Obstacles

Hard vs. Soft Obstacles

What is the cost of collision?



Perhaps it is an optimization task, as opposed to feasibility.

Other Tasks

Obstacle avoidance might not be the primary objective.

- Manipulate objects in the environment
- Rendezvous with other robots
- Maintain visibility of moving obstacles
- Search and destroy moving obstacles
- Count the number of moving obstacles

Can imagine a relation over pairs of bodies: collide, not collide, see, not see, and so on.

Plan Representations

Off-Line vs. On-Line Planning

If obstacles are not completely predictable, then sensors may provide improved information during execution.

Off-line: The plan is computed before the robot is placed into the environment. There is time to plan carefully.

On-line: A plan is requested during execution, based on new information. Planning time is critical.

Dynamic replanning
Anytime planning
Partial planning
Receding horizon control

Plan Representations

1. Time feedback
2. Configuration/state feedback
3. Receding horizon model
4. Information feedback

This plan specifies the configuration $q(t)$ at every time $t \in T$:

$$\pi : T \to Z_{free}$$

Problems with π:

- Need to determine obstacle reachable sets to obtain Z_{free}.
- What if the robot cannot be perfectly controlled?
- What if new information about obstacles is sensed?

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Configuration Feedback

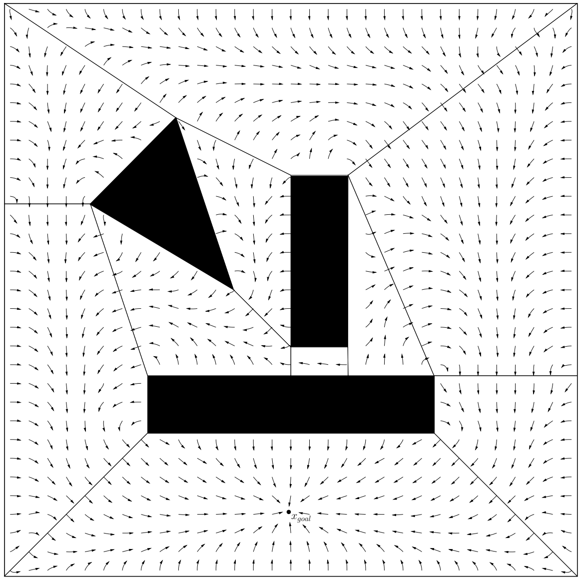
Specify a velocity or action at every configuration:

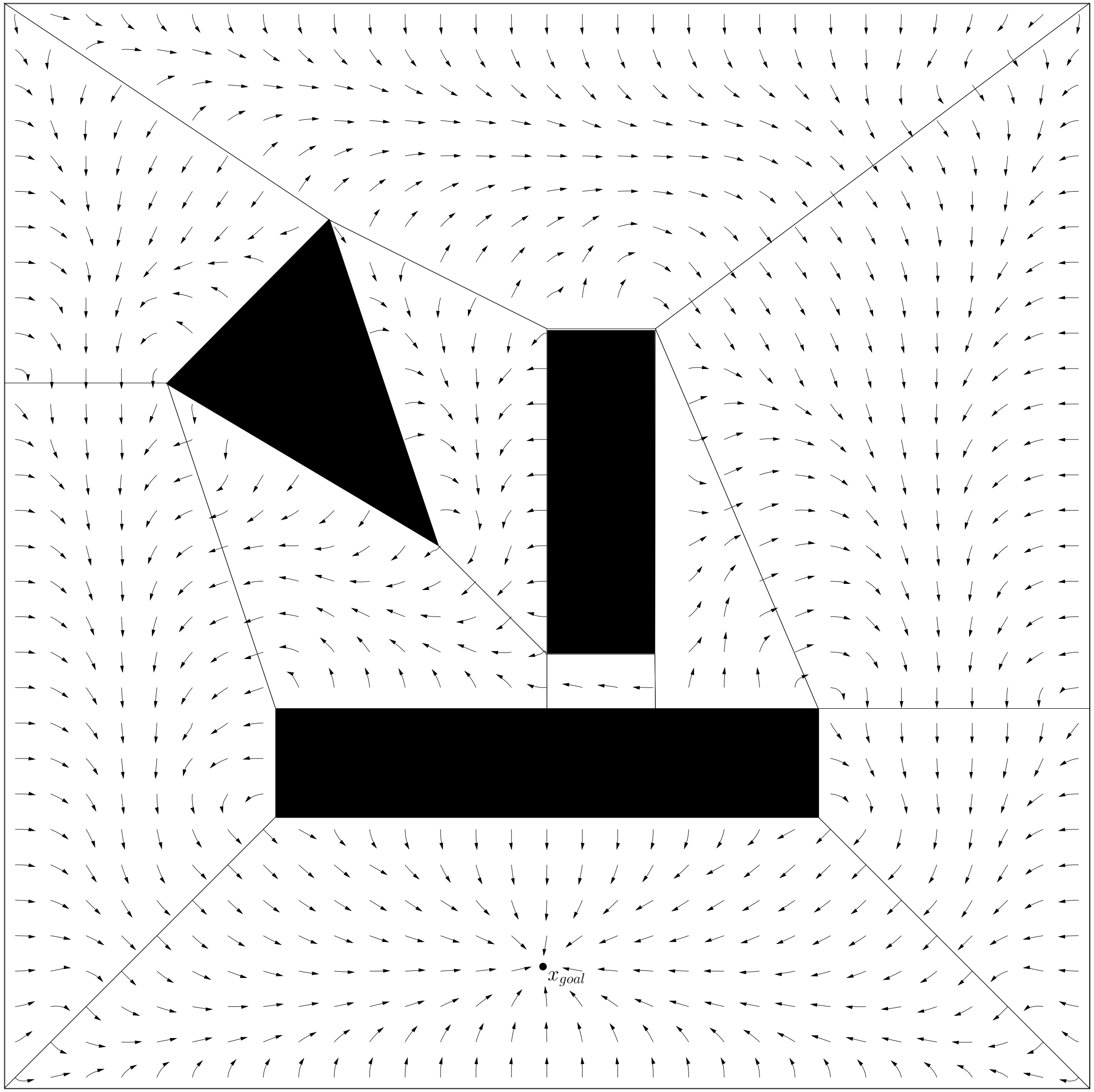
$$\pi : \mathcal{C}_{free} \to U$$

Could obtain $u = \pi(q)$ as the gradient of an energy function.
Potential functions, navigation functions, Lyapunov functions

Possible systems: $\dot{q} = u$ or $\dot{q} = f(q, u)$.

Compute a collision-free vector field:





x_{goal}

State Feedback

Moving into the phase space is common.
$x = (q, \dot{q})$

$\ddot{q} = u$ or $\dot{x} = f(x, u)$.

A state-feedback control law:

$$\pi : X \to U$$

Note that the state must be accurately sensed at all times.
This includes obstacle velocities.

Maybe a Kalman filter can be used.

Dynamic replanning; receding horizon control; model predictive control.

Consider an infinite horizon problem.
Transition model: $x_{k+1} = f(x_k, u_k)$.
Cost functional:

$$\sum_{i=1}^{\infty} l(x_i)$$

Example:

$$l(x_i) = \begin{cases} 0 & \text{if } x_i \in X_G \\ \infty & \text{if } x_i \in X_{obs} \\ d(x_i) & \text{otherwise.} \end{cases}$$

in which $d(x_i)$ is an underestimate of the distance (number of steps) to X_G.

Key difficulty: X_{obs} may change in the future.

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Consider a window size of n steps.

Let k be the current stage.

Choose $(u_k^*, u_{k+1}^*, \ldots, u_{k+n}^*)$ to optimize

$$\sum_{i=k}^{k+n} l(x_i)$$

Apply u_k^* to obtain $x_{k+1} = f(x_k, u_k^*)$.

Dynamic Replanning

Consider a window size of n steps.

Let k be the current stage.

Choose $(u^*_k, u^*_{k+1}, \ldots, u^*_{k+n})$ to optimize

$$\sum_{i=k}^{k+n} l(x_i)$$

Apply u^*_k to obtain $x_{k+1} = f(x_k, u^*_k)$.

X_{obs} may have changed.

Choose $(u^*_{k+1}, u^*_{k+2}, \ldots, u^*_{k+n+1})$ to optimize

$$\sum_{i=k+1}^{k+n+1} l(x_i)$$

Apply u^*_{k+1} to obtain $x_{k+2} = f(x_{k+1}, u^*_{k+1})$.

Dynamic Replanning

Consider a window size of n steps.

Let k be the current stage.

Choose $(u_k^*, u_{k+1}^*, \ldots, u_{k+n}^*)$ to optimize

$$\sum_{i=k}^{k+n} l(x_i)$$

Apply u_k^* to obtain $x_{k+1} = f(x_k, u_k^*)$.

X_{obs} may have changed.

Choose $(u_{k+1}^*, u_{k+2}^*, \ldots, u_{k+n+1}^*)$ to optimize

$$\sum_{i=k+1}^{k+n+1} l(x_i)$$

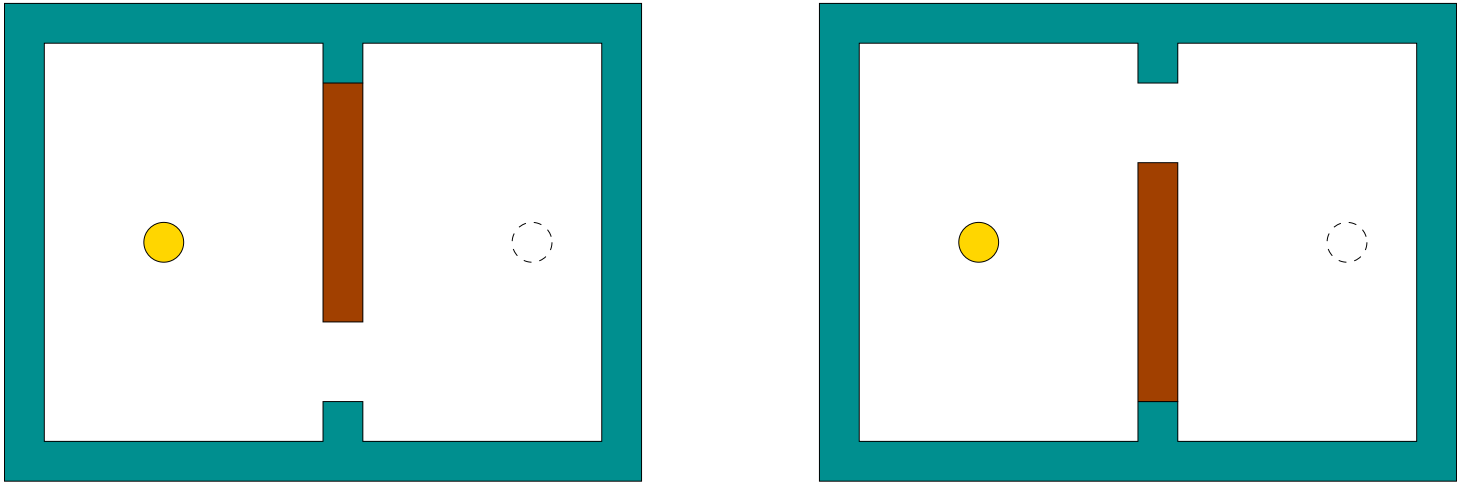
Apply u_{k+1}^* to obtain $x_{k+2} = f(x_{k+1}, u_{k+1}^*)$.

X_{obs} may have changed.

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Questions:

- Does this lead to globally optimal solutions?
- Does this even produce stable robot motions?



Mode 1 Mode 2

- In many (most?) settings, the obstacle region $\mathcal{O}(t)$ is not completely known.
- Even geometric representations of portions may be difficult.
- Also, is the robot configuration always known?

- Information comes from sensors, before and during execution.
- What representations can be reliably, efficiently maintained?
- Given the task, what representations are appropriate? What information is necessary?
- Can performance guarantees be made in spite of missing information?

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

A Common Structure

- Localization only: Set of possible configurations
- Mapping only: Set of possible environments
- Both: Set of configuration-environment pairs

Let $\mathcal{Z}$ be any set of sets.

Each $Z \in \mathcal{Z}$ is a "map" .
Each $z \in Z$ is the configuration or "place" in the map.

Unknown configuration and map yields a state space as:
All (z, Z) such that $z \in Z$ and $Z \in \mathcal{Z}$.

Sensors Lose Information

There is a giant state space: $X \subset \mathcal{C} \times \mathcal{E}$

E is the set of all *environments*.

Given a set of k possible maps:

$$\mathcal{E} = \{E_1, E_2, \ldots, E_k\}$$

For example, could be given 5 maps:

$$\mathcal{E} = \{E_1, E_2, E_3, E_4, E_5\}$$

X is all (q, E_i) in which $(q_x, q_y) \in E_i$ and $E_i \in \mathcal{E}$.

Recall the common structure.

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

State Space For Unknown Map

Given an infinite *map family*, $\mathcal{E}$, of environments.

Examples:

- The set of all connected, bounded polygonal subsets that have no interior holes (formally, they are *simply connected*).
- The previous set expanded to include all cases in which the polygonal region has a finite number of polygonal holes.
- All subsets of $\mathbb{R}^2$ that have a finite number of points removed.
- All subsets of $\mathbb{R}^2$ that can be obtained by removing a finite collection of nonoverlapping discs.
- All subsets of $\mathbb{R}^2$ obtained by removing a finite collection of nonoverlapping convex sets.
- A collection of piecewise-analytic subsets of $\mathbb{R}^2$.
- The set of all bitmap representations.

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Sensing Model

There is a giant state space: $X \subset \mathcal{C} \times \mathcal{E}$

Let Y be an *observation space*
A *sensor mapping* is:

$$h : X \to Y$$

When $x \in X$, the sensor instantaneously observes $y = h(x) \in Y$.

Could make a noisy sensor version: $p(y|x)$

We might even want the state-time space: $Z = X \times T$

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

The amount of state uncertainty due to a sensor

$$h : X \to Y$$

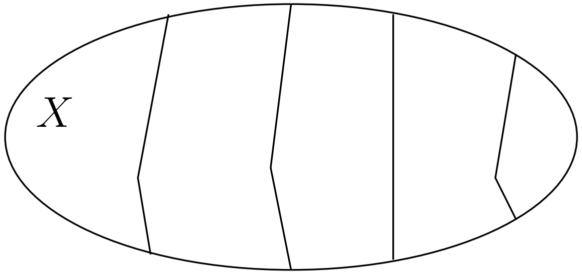
The preimage of an observation y is

$$h^{-1}(y) = \{x \in X \mid y = h(x)\}$$

Think about the uncertainty being handled here!

Suppose X and $h : X \rightarrow Y$ are given.

The set of all preimages partitions X

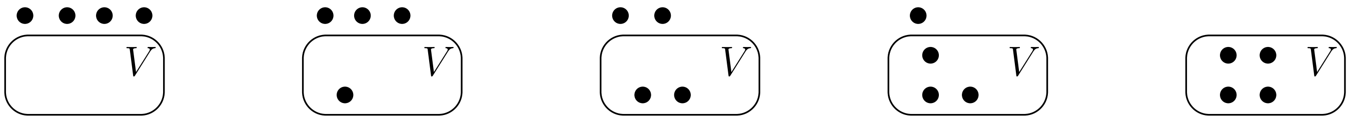


There is one preimage for every $y \in Y$.

Let $\Pi(h)$ be the partition X that is induced by h.

Detection Sensor Example of $\Pi(h)$

- n point bodies move in $\mathbb{R}^2$.
- $X = \mathbb{R}^{2n}$
- $Y = \{0, 1, \ldots, n\}$
- The sensor mapping $h : X \to Y$ counts how many points lie a fixed detection region V.



For $n = 4$, there are 5 equivalence classes in $\Pi(h)$.

Sensor Feedback

A plan could be represented as

$$\pi : Y \to U$$

Choose the action based only the sensor reading.

Think about swarms, distributed robots, emergent behavior.

Behavior-based robotics.

If sensor feedback is sufficient for the task, great!

If not, then how much memory is needed? How to encode?

What is there to remember?

Action history:

$$\tilde{u}_k = (u_1, \ldots, u_k)$$

Sensor observation history:

$$\tilde{y}_k = (y_1, \ldots, y_k)$$

The Structure of Temporal Filters

Let $\mathcal{I}$ be any set, and call it an *information space*.

Let ι_0 be called the *initial I-state*.

Transition function (filter):

$$\iota_k = \phi(\iota_{k-1}, y_k, u_{k-1})$$

Using the filtering, the system "lives" in $\mathcal{I}$.
The entire world is not reconstructed (unless it is needed).

Information Space Examples

An *information-feedback* plan:

$$\pi : \mathcal{I} \rightarrow U$$

State feedback: I-space is $\mathcal{I} = X$ and plan is $\pi : X \rightarrow U$

Open loop: $\mathcal{I} = \mathbb{N}$ and $\pi : \mathbb{N} \rightarrow U$
π can be written as $(u_1, u_2, u_3, \ldots)$

Sensor feedback: $\mathcal{I} = Y$ and $\pi : Y \rightarrow U$

History feedback: $\mathcal{I} = \mathcal{I}_{hist}$ and $\pi : \mathcal{I}_{hist} \rightarrow U$

Information Space Examples

Belief feedback: $\mathcal{I} = \mathcal{I}_{prob}$ and $\pi : \mathcal{I}_{prob} \rightarrow U$

Each element of $\mathcal{I}_{prob}$ is a pdf of the form $p(x_k | \tilde{u}_{k-1}, \tilde{y}_k)$.

Transitions are accomplished by Bayesian filtering.

Often called Partially Observable Markov Decision Processes (POMDPs).

Set feedback: $\mathcal{I} = \mathcal{I}_{ndet}$ and $\pi : \mathcal{I}_{ndet} \rightarrow U$

Each element of $\mathcal{I}_{ndet}$ is a set of the form $F(\tilde{u}_{k-1}, \tilde{y}_k) \subseteq X$.

Transitions are accomplished by nondeterministic filtering.

More possibilities for $\mathcal{I}$:

- A space of location-topological map pairs
- Relative coordinates of a target to be tracked
- A space of gap navigation trees
- A space of configuration-multiresolution bitmap pairs

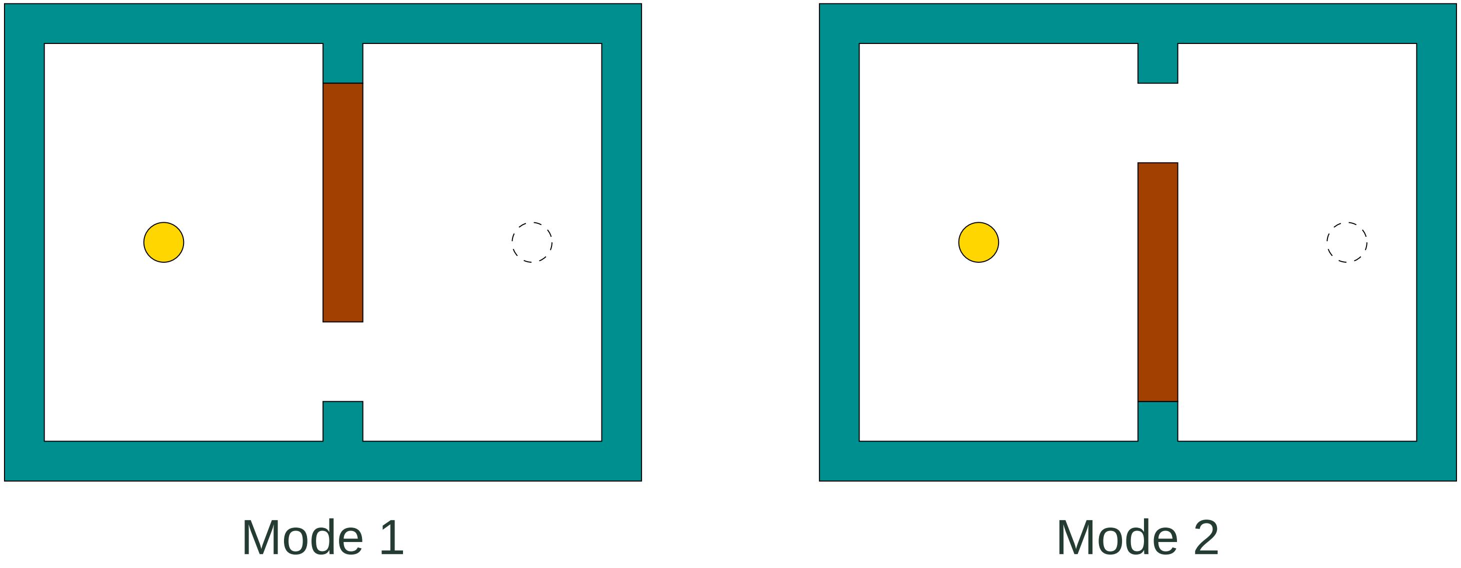
Overall Process

Based on the task, an overall approach that leads to planning:

1. Design the system, which includes the environment, bodies, and sensors.
2. Define the models, which provide the state space X, the sensor mapping h, and the state transition function f.
3. Select an I-space $\mathcal{I}$ for which a filter ϕ can be practically computed.
4. Take the desired goal, expressed over X, and convert it into an expression over $\mathcal{I}$.
5. Compute a plan π over $\mathcal{I}$ that achieves the goal in terms of $\mathcal{I}$.

Really, all steps should be considered together.

Fundamental Limitations



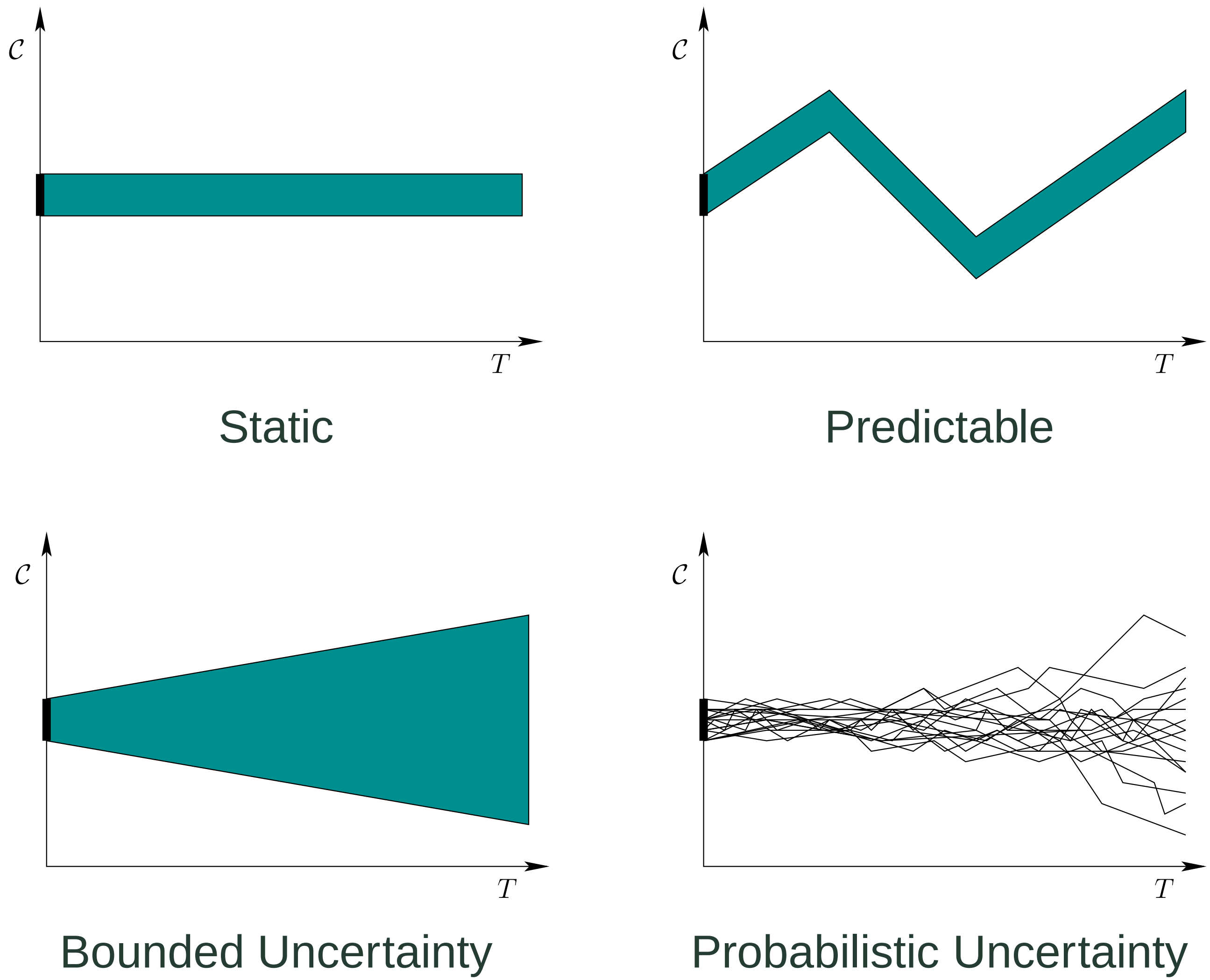
- There is no solution if an adversary moves the wall.
- Receding horizon model leads to oscillation.
- If wall follows a Markov chain, then wait by either potential opening.

What Constitutes a Good Solution?

- Works in a demo
- Works in a simulated demo
- Works robustly, reliably in a deployed system
- Theorems imply that it works for the model
- Both theorems imply correctness and it works in a system

Feasibility vs. optimality

Forward Projections

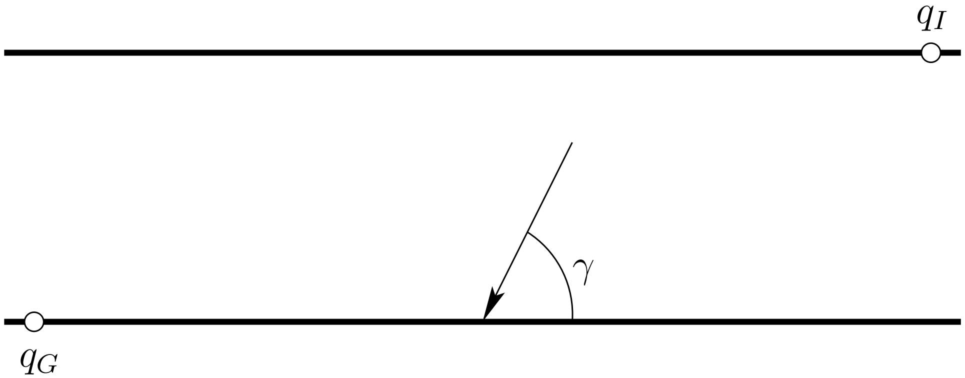


Summary of Part III

- Calculating forward projections (forecasts)
- Do other bodies respond to your motion?
- Hard vs. soft obstacles, other interactions?
- Limitations of replanning
- What kind of information feedback?

Homework 3: Solve During Coffee Break

How should the chicken cross the road?



Environment has two modes: SAFE and DANGER

The chicken moves at constant speed.

Initially SAFE, but DANGER may occur with Poisson arrival parameter λ.

What strategy minimizes the *expected* distance traveled by the chicken?