AMIRKABIR WINTER SCHOOL

Minimalism in Robotics:

From Sensing to Filtering to Planning

PART 5: PLANNING IN INFORMATION SPACES

Steven M. LaValle

March 5, 2012

Overview of Topics

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

1. An I-space view of planning, starting with temporal filters
2. General planning issues
3. Maze searching
4. Visibilty-based pursuit-evasion
5. Shadow information spaces
6. Gap navigation trees
7. Landmark-based navigation
8. Bug algorithms
9. Sensorless manipulation
10. Wild bodies

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

From filters to planning

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Let $\mathcal{I}$ be any I-space.

Assume a filter

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

is given.

Let $G \subset \mathcal{I}$ be a *goal region*.

Starting from ι_0, what sequence of actions u_1, u_2, . . ., will lead to some future I-state $\iota_k \in G$?

Using Filters For Planning

From filters to planning

General issues

Visibility-based pursuit evasion

Maze searching

Gap navigation trees

Learning convex hulls of landmarks

Bug algorithms

Sensorless manipulation

Controlling Wild Bodies

The future may be unpredictable.

Introduce an *I-state dependent* plan:

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

Using a filter ϕ, the execution of a plan can be expressed as

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

The I-space $\mathcal{I}$ is just a sort of “C-space” that is being explored.

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

General issues

Main Issues

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

The following issues arise repreatedly in planning:

1. **Predictability**
2. **Reachability**
3. **Optimality**
4. **Computability**

Predictability

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Are the effects of actions predictable in the I-space $\mathcal{I}$?

If **yes**, then a *path* through the I-space is obtained.

Example: Sensorless manipulation

Example: Visibility-based pursuit evasion

By analogy to path planning in C-space:

1. Combinatorial planning in I-space
2. Sampling-based planing in I-space

If **no**, then information feedback is critical

It is like feedback planning (or control) in C-space, but instead over I-space

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Reachability:
Is the goal region $G \subset \mathcal{I}$ even reachable from the initial I-state?

Do there even exist actions that will take us to G?

Does there exist a plan that can reach G?

With unpredictability, is G *guaranteed* to be reached, over all possible disturbances?

A more basic question is whether the goal can even be adequately expressed in $\mathcal{I}$.

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Perhaps many plans can reach G

What criteria should be formulated to compare plans?

Which plans are the best, or optimal with respect to criteria?

Do optimal plans even exist?

Computability

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Given a description of the problem, can an algorithm be determined that automatically computes a useful plan?

Sometimes a clever human designs the plan (e.g. bug algorithms)

What is the algorithmic complexity of computing a solution plan?

What is the implementation difficulty of computing a solution plan?

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Important Generic Examples

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

Recall the previous filters over these 4 I-spaces.

Now we move from *passive* to *active*.

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Overall Process

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

1. Design the system, which includes the environment, bodies, and sensors.

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

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.

Overall Process

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

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.

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Overall Process

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

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 $\mathcal{I}$.

Overall Process

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

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.

Might have to backtrack.

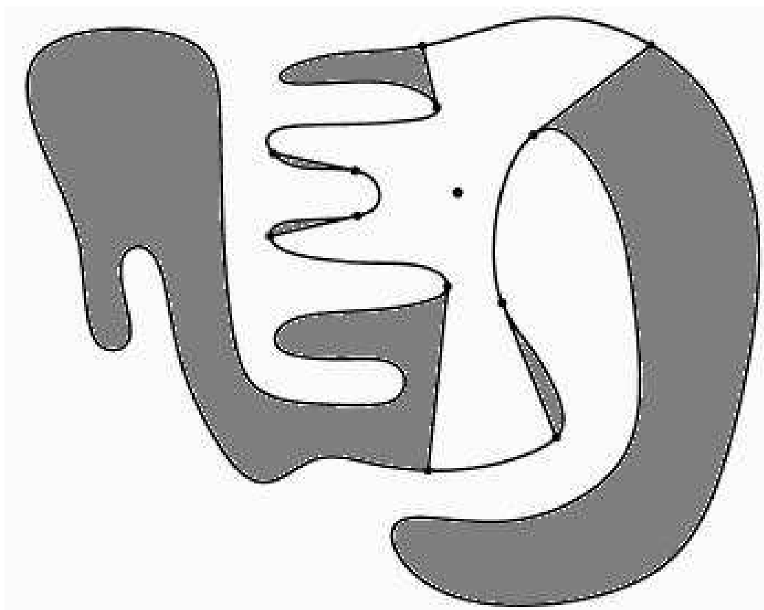
From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

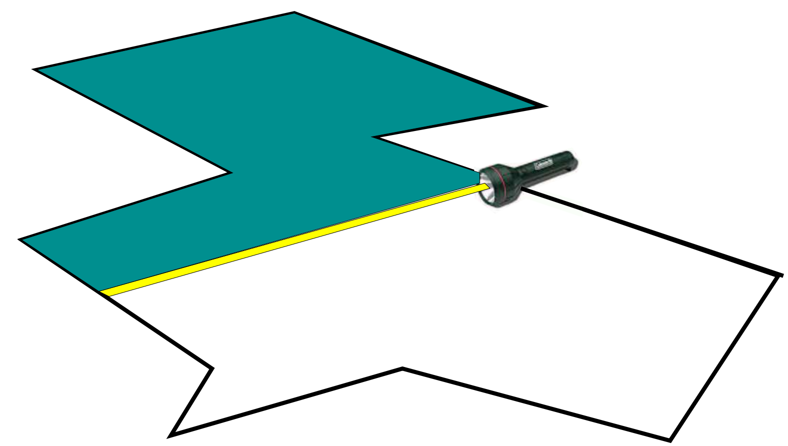
Visibility-based pursuit evasion

Model

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

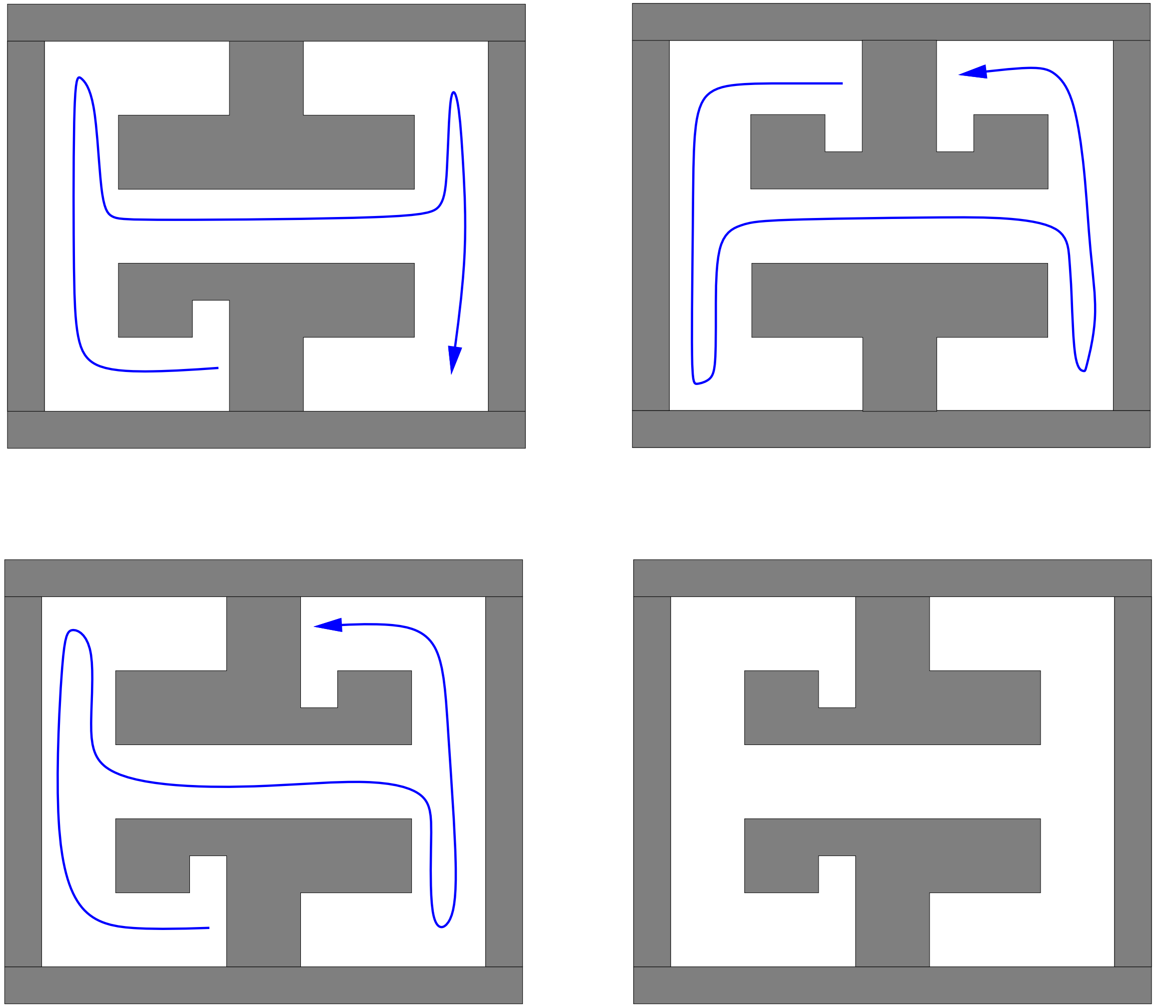
- A 2D environment, possibly curved
- Unpredictable point “evaders” move with unbounded speed
- Point “pursuers” use visibility sensors to find all evaders





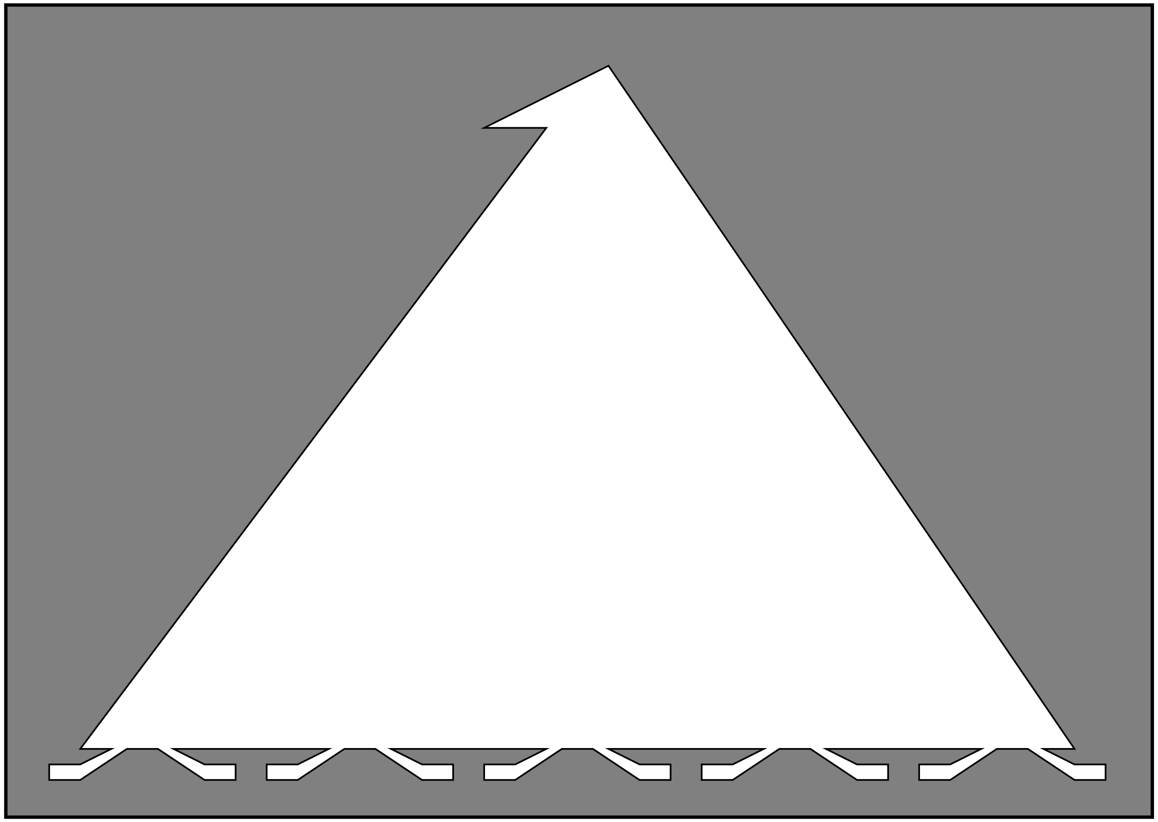
From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Try to solve using one pursuer with 360° vision:



From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

You might have to revisit the same place many times...



$\Omega(n)$ recontaminations

A Cell Decomposition

From filters to planning

General issues

Visibility-based pursuit evasion

Maze searching

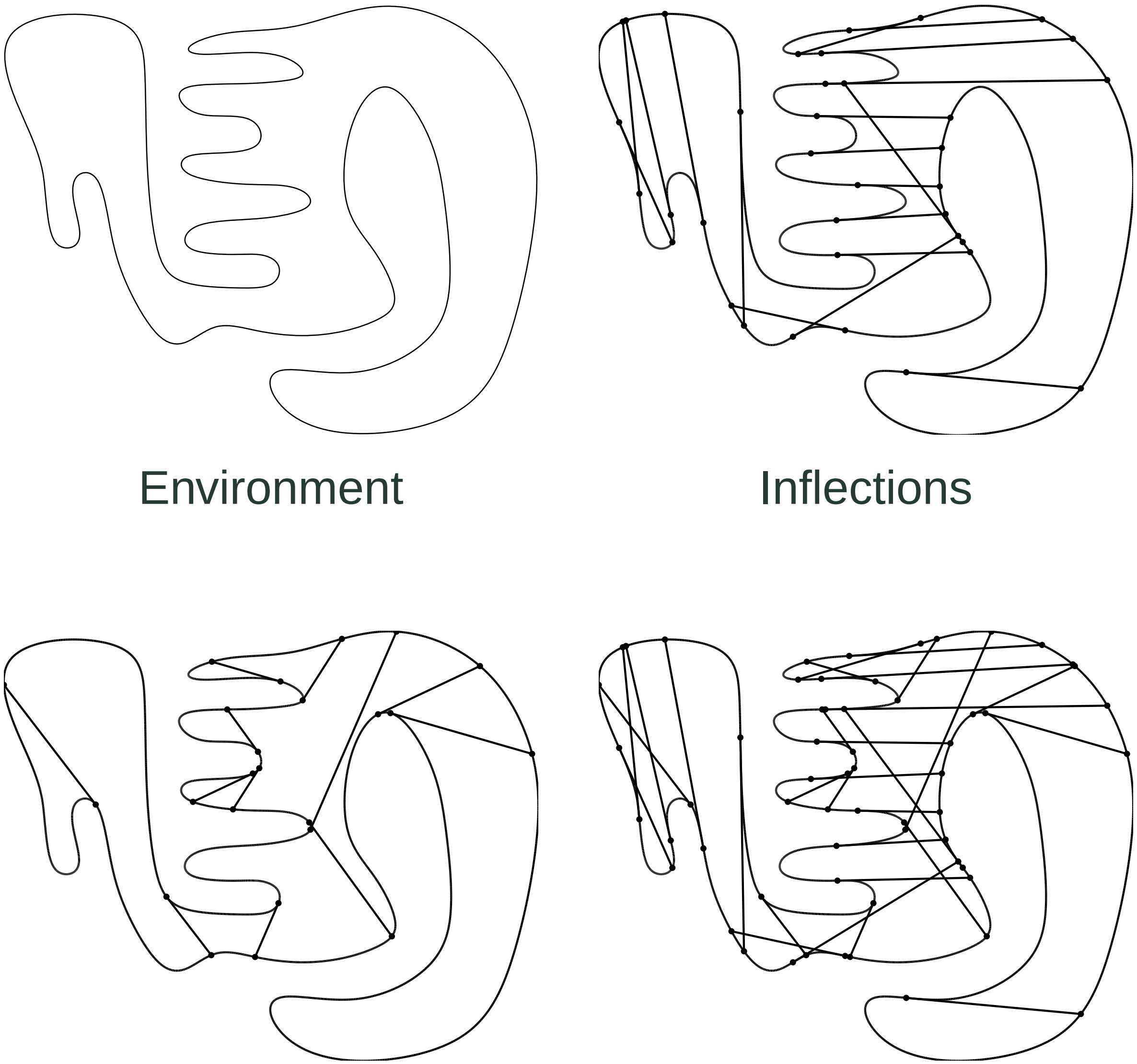
Gap navigation trees

Learning convex hulls of landmarks

Bug algorithms

Sensorless manipulation

Controlling Wild Bodies



Environment

Inflections

Bitangents

Cell Decomposition

The Information Space

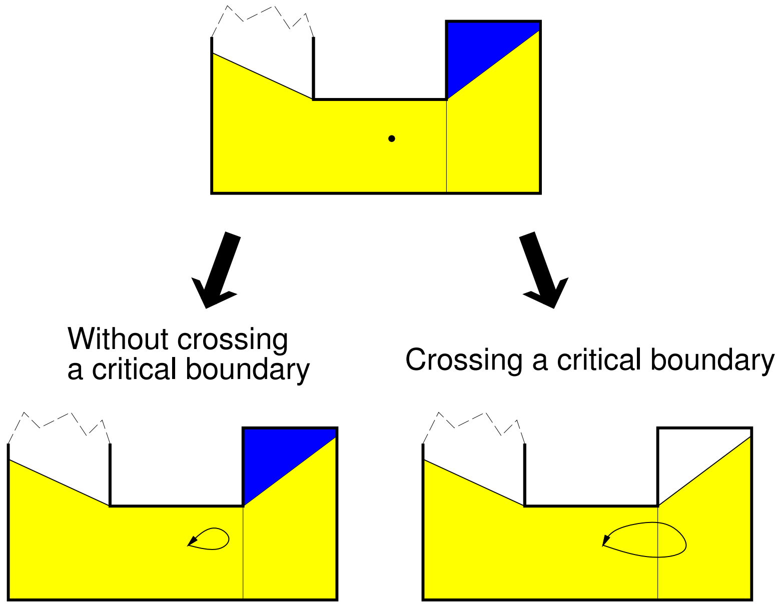
Identify all unique situations that can occur:

An information state is identified by (x, S) in which

x = *the position of the pursuer*

S = *set of possible evader positions*

The set of all information states forms an information space.



Many closed-path motions retain the same information state.

Systematic Graph Search

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

- Let $G(V, E)$ be the dual of the cell decomposition
- For each $v \in V$, there are finitely many information classes
- Form a directed information state graph, $G_I(V_I, E_I)$
- Each $v \in V_I$ is an information class
- Each $e \in E_I$ indicates a transition between information classes (crossing an inflection or bitangent)

For each information class, label each shadow component with “1” for *contaminated* or “0” for *clear*.

Search G_I from a state in which

All labels are “1”

to a state in which

All labels are “0”

From filters to planning

General issues

Visibility-based pursuit evasion

Maze searching

Gap navigation trees

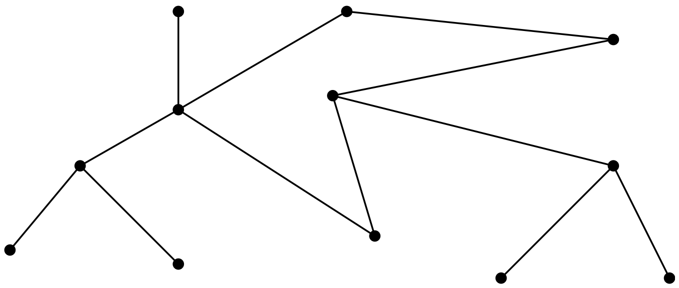
Learning convex hulls of landmarks

Bug algorithms

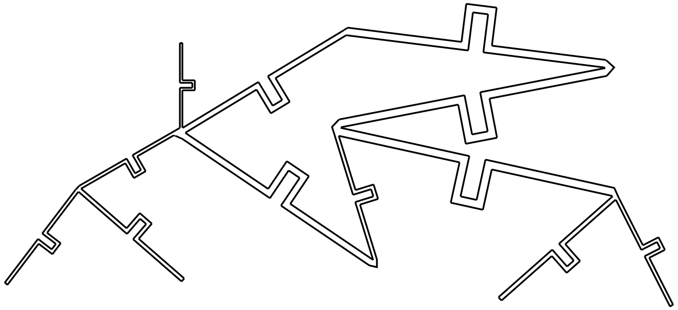
Sensorless manipulation

Controlling Wild Bodies

Pursuit-evasion planar graph problem:



A geometric equivalent:



Results:

Deciding whether a simple polygon can be searched by k pursuers is NP hard.

$\Omega(\lg n + \sqrt{h})$ pursuers needed for some polygons

From filters to planning

General issues

Visibility-based pursuit evasion

Maze searching

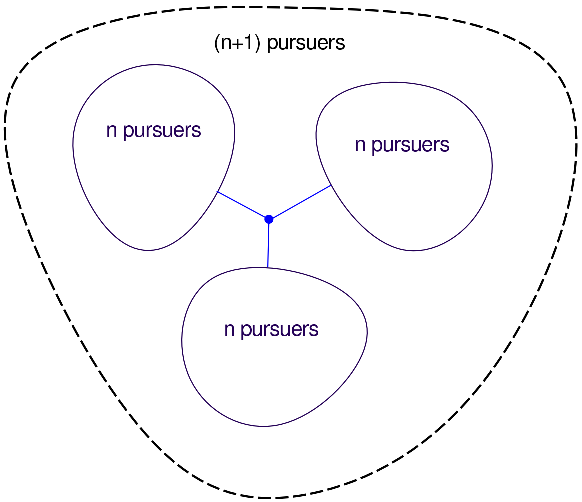
Gap navigation trees

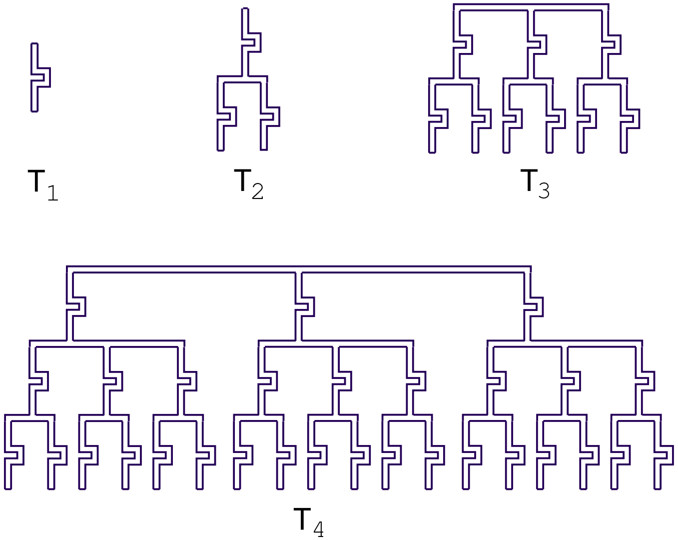
Learning convex hulls of landmarks

Bug algorithms

Sensorless manipulation

Controlling Wild Bodies





This sequence requires $\Omega(\lg n)$ pursuers.

Pursuit-Evasion Results with Perfect Models

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

- Constructing and searching equivalence classes in the information space
- A complete algorithm for 360° visibility
- A complete algorithm for 1 pursuer with 1 flashlight
- A complete algorithm for 2 pursuers with 1 flashlight each

All of these assume perfect mapping, control, and localization.

Alternative pursuit-evasion approaches:

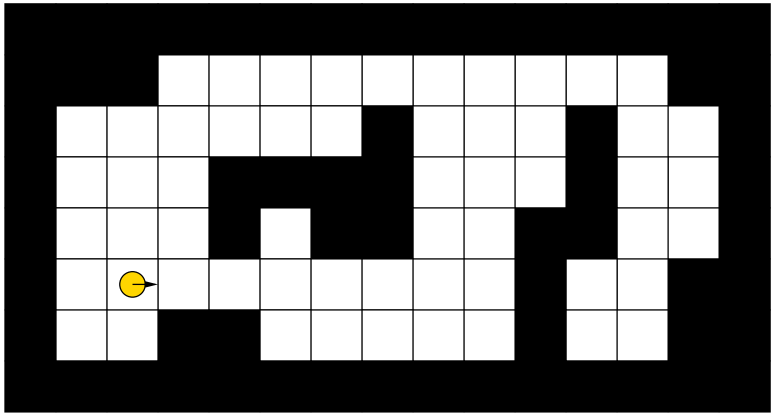
- Using the gap sensor (Sachs, Rajko, LaValle, IJRR 2004)
- Using a wall-following robot (Katsev, Tovar, Yershova, Ghrist, LaValle, IEEE Trans. Robotics, 2011

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Maze searching

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies



Each $E \in \mathcal{E}$ is a bounded set of white tiles.

$$X \subset \mathbb{Z} \times \mathbb{Z} \times D \times \mathcal{E}$$

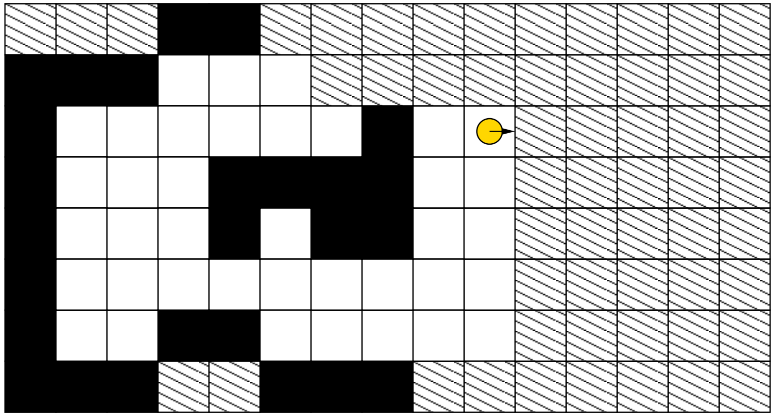
Actions: 1) Rotate 90 degrees CCW; 2) Move foward one tile.

Task: Make a plan that systematically searches all white tiles.
For example, find a hidden treasure.

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Could try $\mathcal{I} = \mathrm{pow}(\mathbb{Z} \times \mathbb{Z} \times D \times E)$.
Too large!

Instead, maintain I-states B (known black tiles) and W (known white tiles).

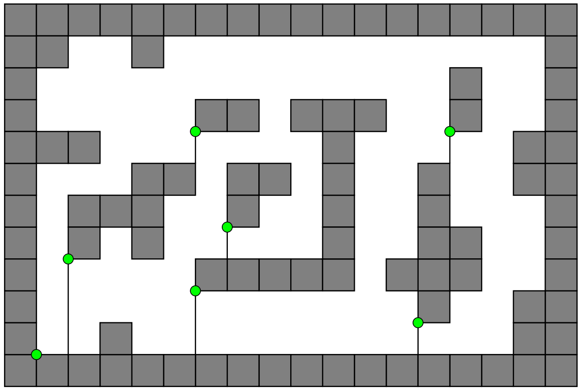


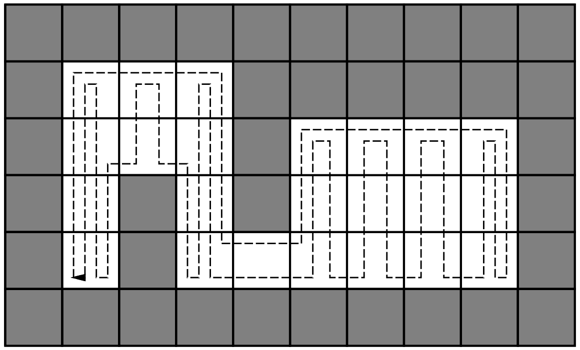
All other tiles assumed "unknown".

I-space $\mathcal{I}$ is all ways to partition $\mathbb{Z} \times \mathbb{Z}$ into connected "white", "black", and "unknown" tiles.
Linear space required for an I-state (filter memory).

Maze Searching: Blum and Kozen 1978

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies





I-state: Latitude (integer) and orientation (two bits)

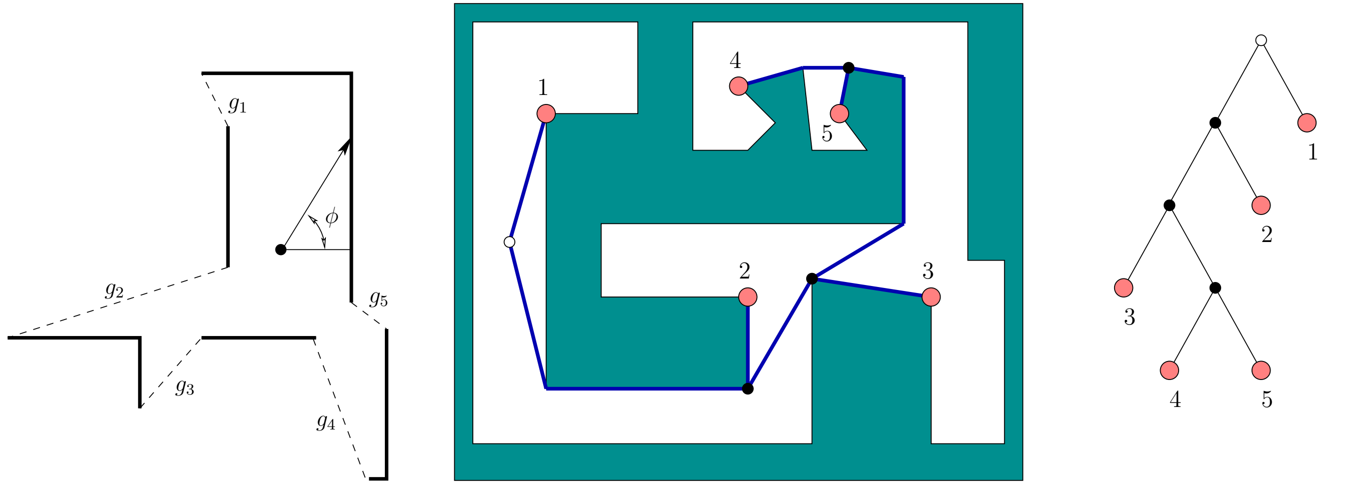
Only logarithmic space required: Not enough for a “map”.

They found an I-space that is much smaller than the set of all maps.

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Gap navigation trees

Make Gap Navigation Trees Active



For gap navigation trees, two active tasks:

1. Full exploration of the environment
2. Distance-optimal navigation to retrieve objects

Tovar, Murrieta, LaValle, *IEEE Trans. Robotics*, 2007.

Introduce an Action

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

A gap-chasing action is introduced:

Move the robot toward a gap g until a critical event occurs.

One of two events must occur:

1. The gap g splits into two gaps g' and g''.
2. The gap g disappears.

Full Exploration

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

If a gap ever appears, mark it as *primitive*.

This is an extension to the filter I-state.

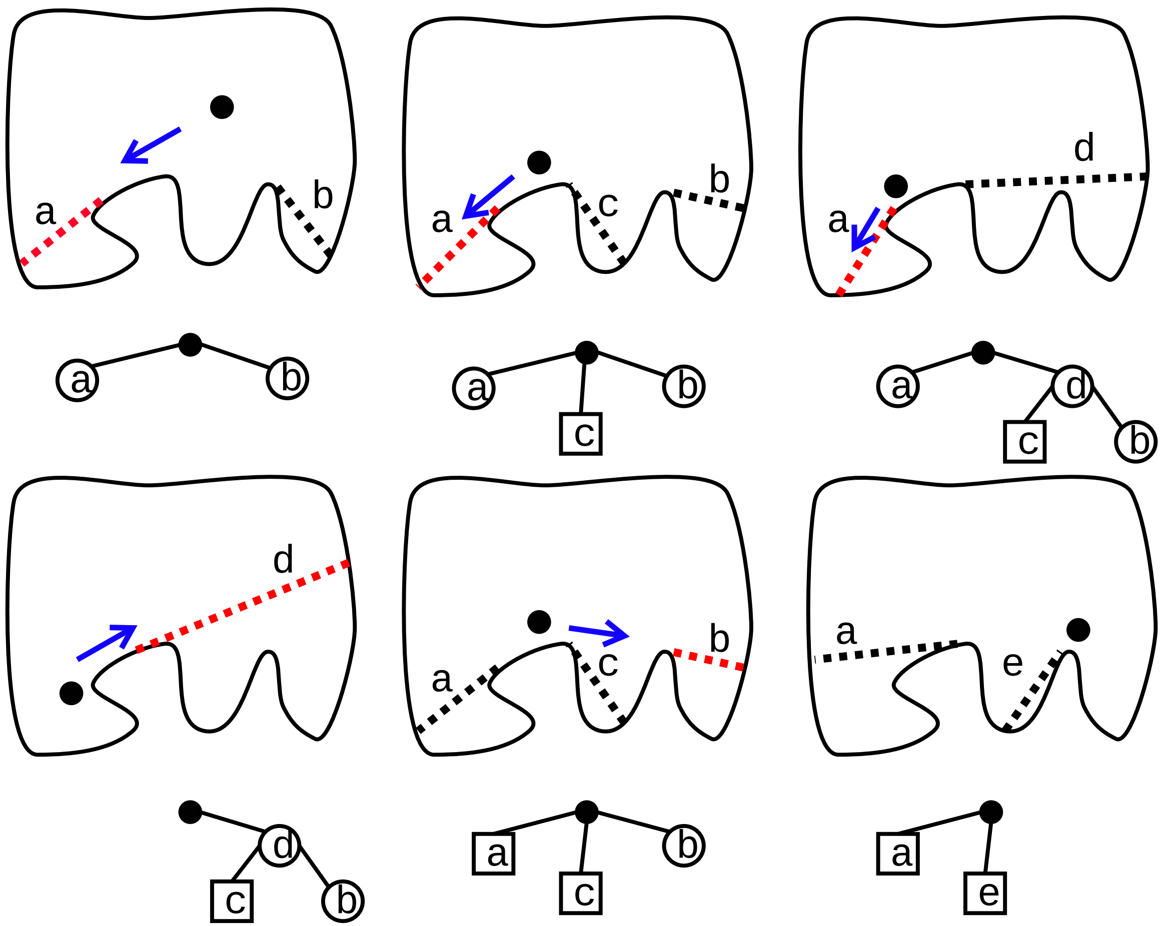
1. Mark all gaps in the initial tree as *non-primitive*.
2. Let $k = 1$.
3. Chase any gap g that is a non-primtive leaf.
4. If g *disappears*, then go to Step 6.
5. If g *splits*, then chase one of its children.
6. Unless all leaves are primitive, increment k and go to Step 3.

At the end, all leaves are primitive and the environment has been fully explored.

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

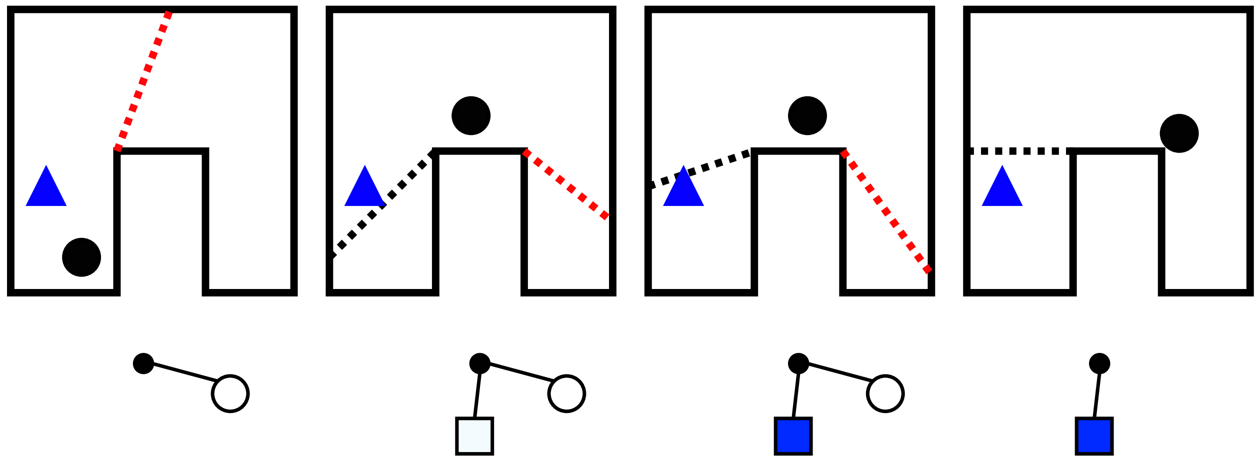
Chase every non-primitive leaf:



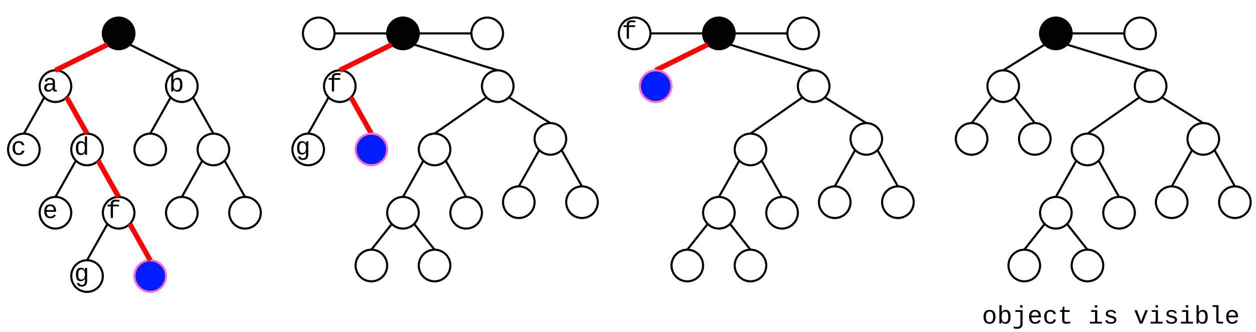
Eventually, all leaves become primitive.

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

There are no coordinates.



Objects hide behind gaps.



Chase the appropriate sequence of gaps.

Possible Current States

From filters to planning

General issues

Visibility-based pursuit evasion

Maze searching

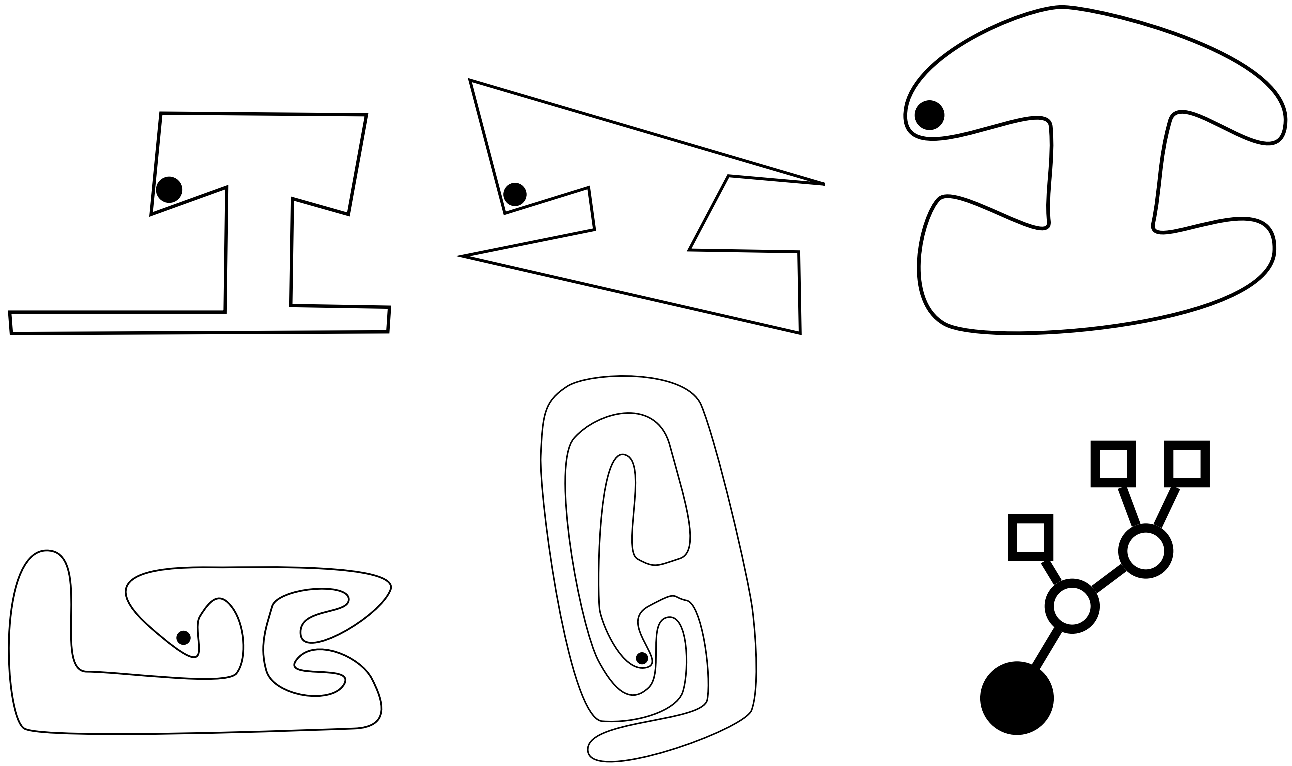
Gap navigation trees

Learning convex hulls of landmarks

Bug algorithms

Sensorless manipulation

Controlling Wild Bodies



Many configuraton-environment pairs have the same tree.

The robot does not have to distinguish!

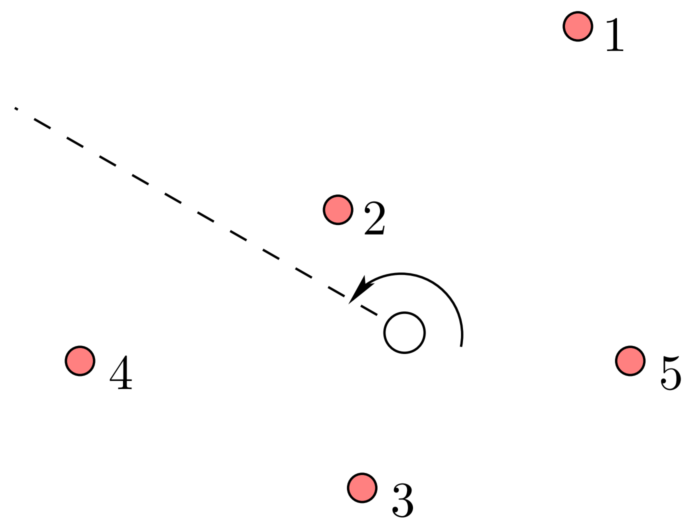
From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Learning convex hulls of landmarks

Recall: Cyclic Permutation Sensor

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Relation: “is to the left of in counterclockwise order”



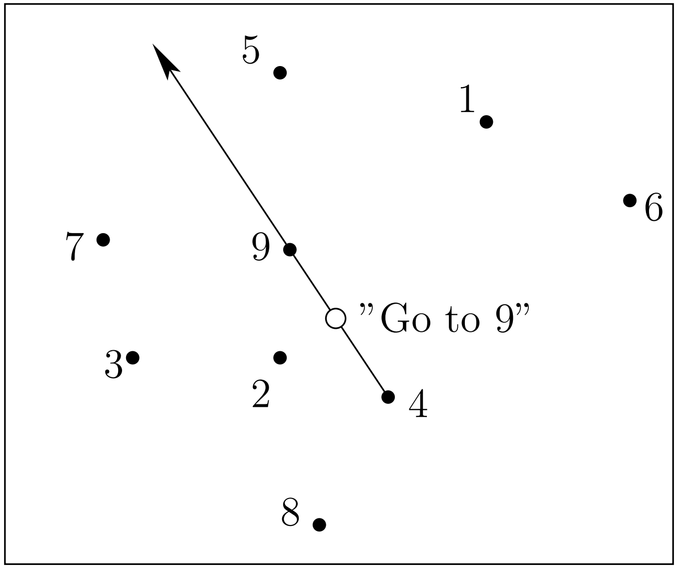
Observation: $y = (1, 2, 4, 3, 5)$

Making an Active Version

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

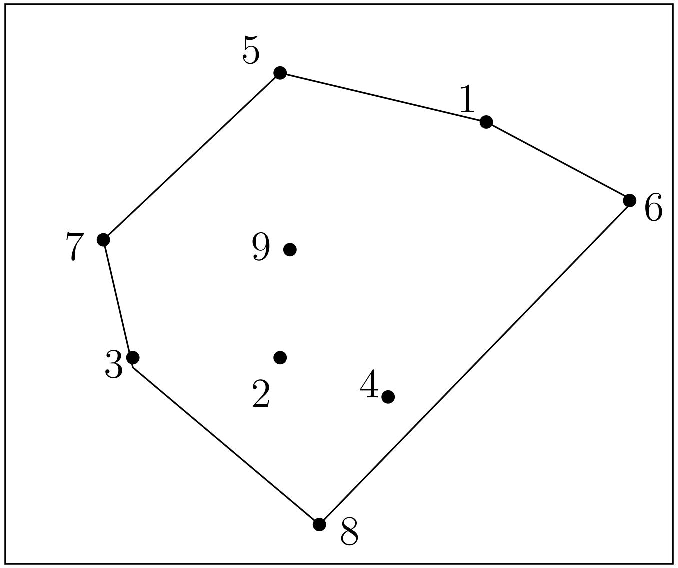
Tovar, Freda, LaValle, 2007.

- Landmark locations are unknown
- Introduce action: “Go to landmark i”
- Can notice which landmarks are “to the left” of the path.



Sense that $(6, 1, 5)$ is to the right of $(7, 2, 3, 8)$.

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies



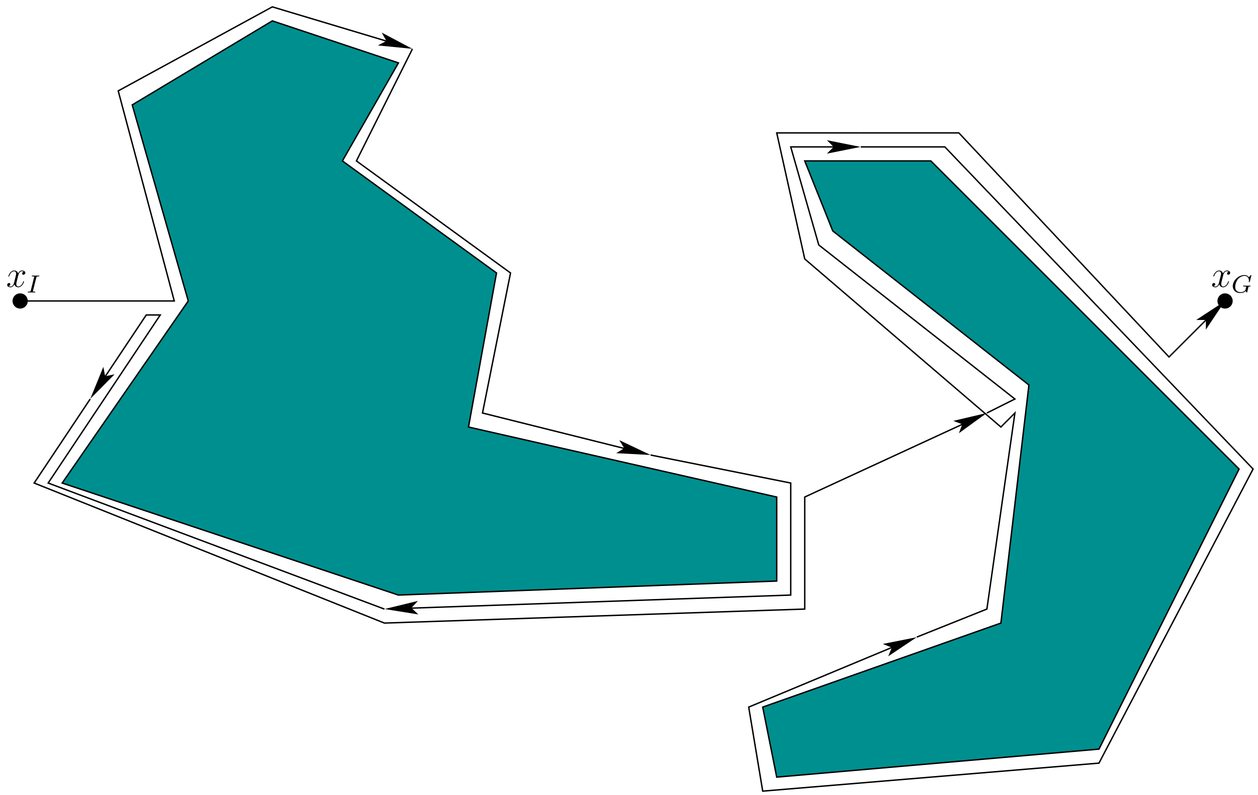
By visiting all pairs, the filter can learn:

- For any subset $L' \subset L$ of landmarks, which others in L lie in the convex hull of L'.
- Equivalently, the robot learns the dual arrangement, order types, oriented matroid.
- The robot can navigation to any goal specified as a cyclic permutation.

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Bug algorithms

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies



Navigate without being given an initial map E

Lumelsky, Stepanov, 1987; Kamon, Rivlin, Rimon, 1997; many others...

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Taylor, LaValle, ICRA 2009

- The plane contains unknown obstacles with piecewise analytic boundary.
- Each obstacle boundary has finite length.
- A *tower* sends a constant signal.
- Robot has very limited sensors.
- Command the robot so that it reaches the tower.

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

- **Boundary (or Contact) sensor:**
  Indicate whether or not robot is on the boundary.
- **Tower alignment/gradient sensor:**
  Indicate whether robot is facing the tower (or intensity gradient).
- **Transformed signal intensity sensor:**
  Observe the value of $m(p - p_t)$.

Regarding m:

- m has only only local maximum, at the tower.
- The function m itself is not given.
- Level sets of m may be symmetric (circles) or asymmetric.

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Actions

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

There are three possible actions:

u_{fwd}: Go straight until either ∂E is hit, tower is hit, or local intensity maximum detected.

u_{fol}: Follow ∂E until local maximum detected.

u_{ori}: Rotate until facing tower (or local gradient).

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Illustration of the Plan

Guaranteed to converge; upper bound on distance shown.

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

State space: $X \subset \mathbb{R}^2 \times S^1 \times \mathcal{E}$

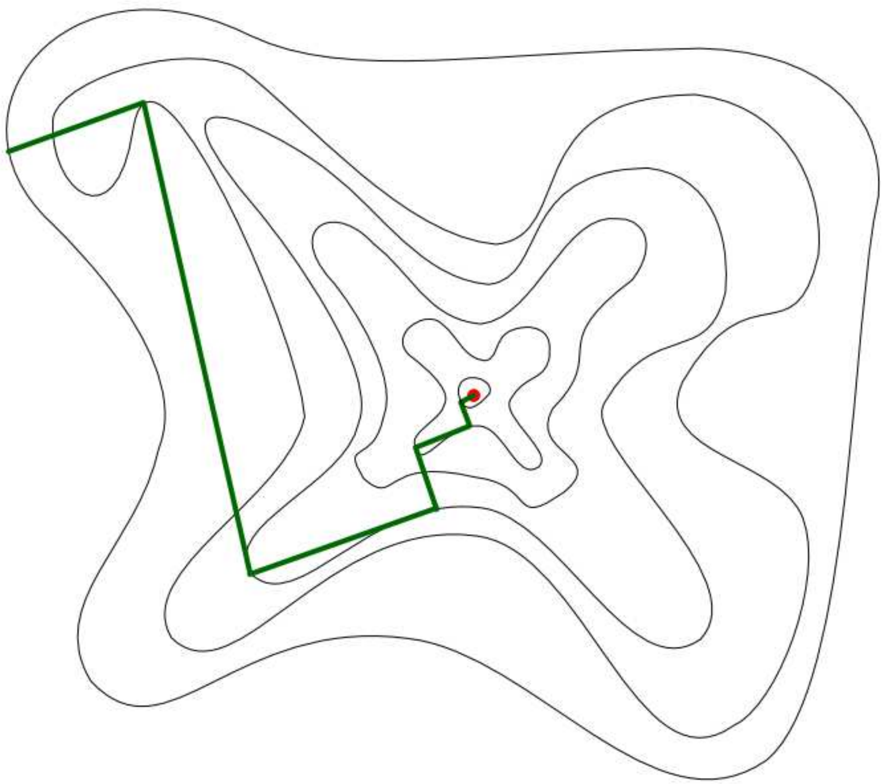
I-space: $\mathcal{I} = Y^3 \subset \mathbb{R}^3$

I-state components:

1. Current observation
2. Observation when obstacle was last contacted
3. Observation just prior to application of u_{fwd}

Multiple Iterations in the Interior

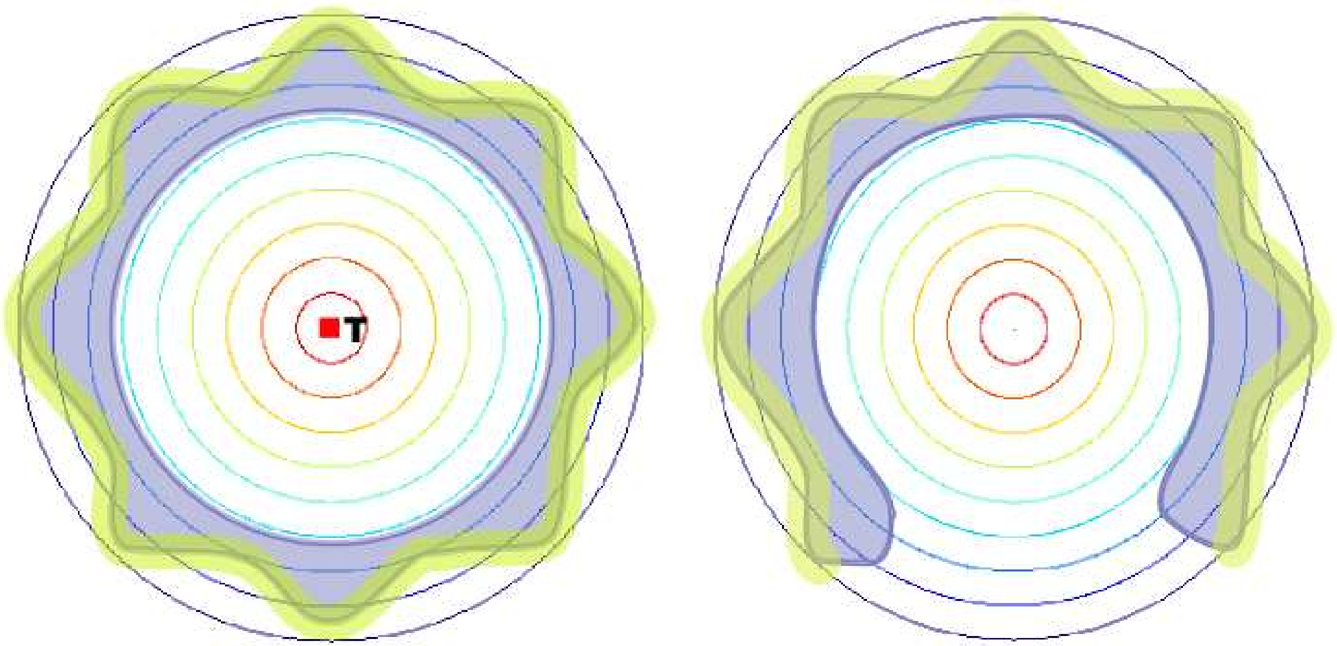
From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies



- Equivalent to Steepest Descent with Line Search.
- Result: Convergence is obtained, but distance bound depends on properties of m.

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Proposition: Using its sensors and motion primitives, it is impossible for the robot to determine whether the tower is reachable, in other words whether $p_t \in E$.

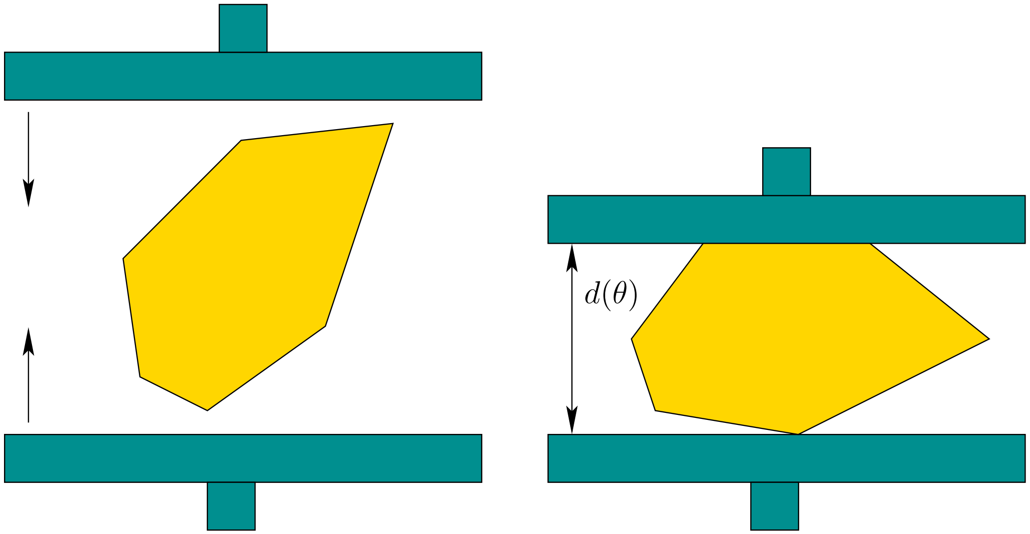


From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Sensorless manipulation

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Try to force parts into a known orientation

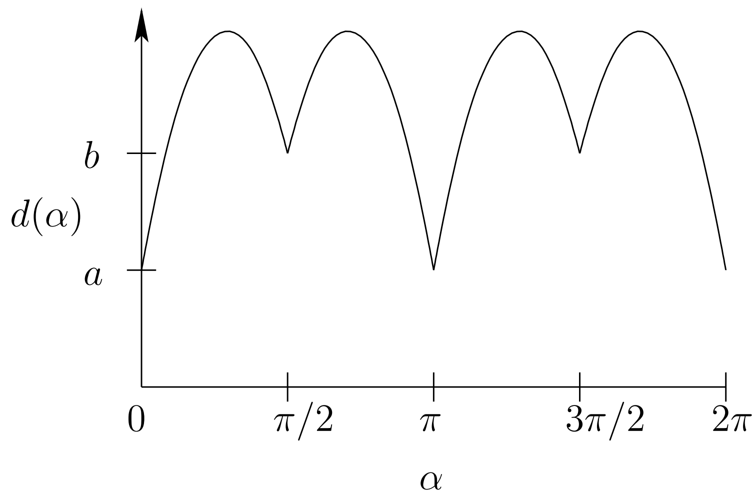


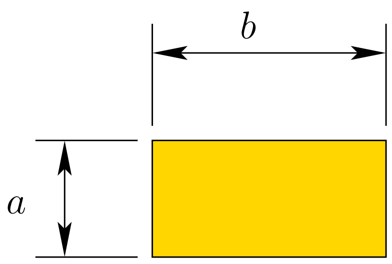
Mason, Goldberg, 1990

$$X = S^1 \quad \mathcal{I} = \mathrm{pow}(X)$$

Plan: $\pi = (u_1, u_2, \ldots, u_n)$
A sequence of squeeze operations

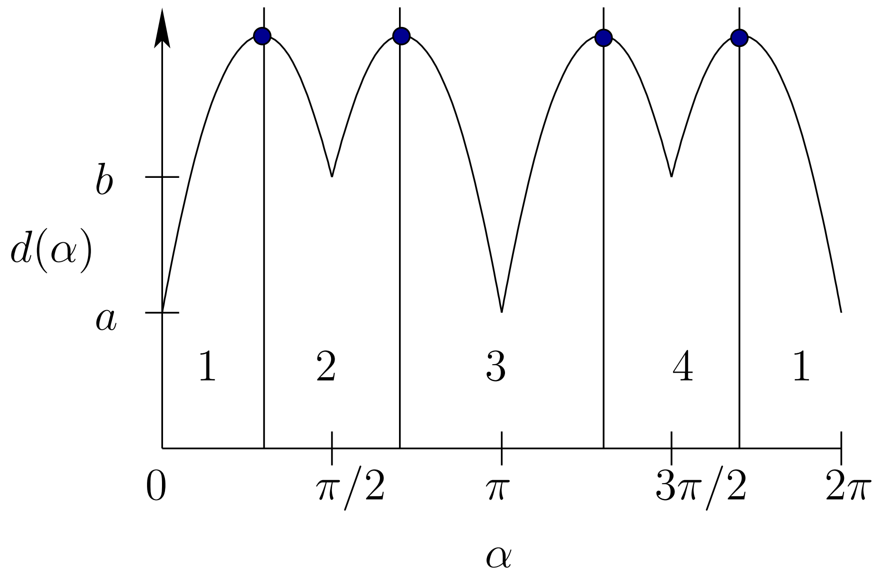
From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies





Consider the “diameter” as a function of orientation.

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies



There are four regions of attraction.

This causes a funneling effect when actions are applied.

Sensorless Manipulation

From filters to planning

General issues

Visibility-based pursuit evasion

Maze searching

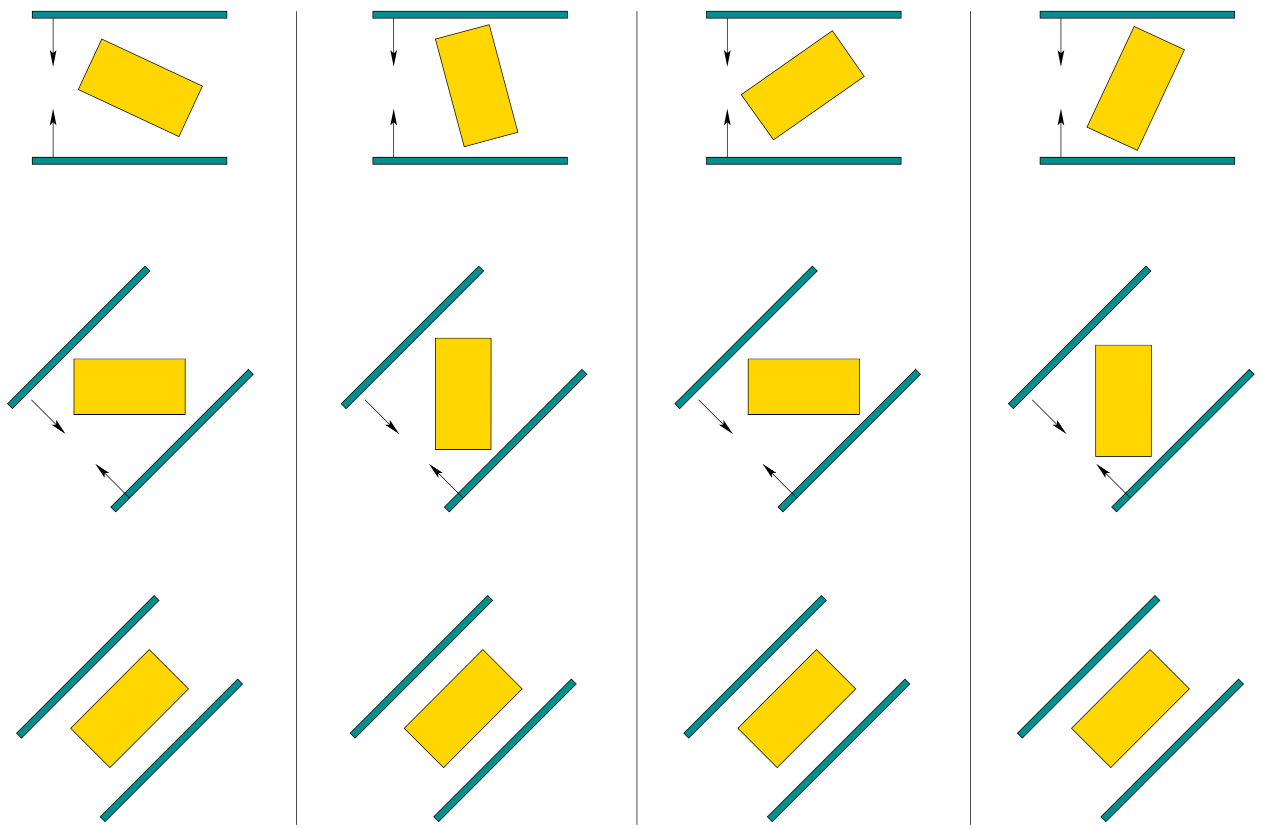
Gap navigation trees

Learning convex hulls of landmarks

Bug algorithms

Sensorless manipulation

Controlling Wild Bodies



A computed plan that applies two squeeze actions

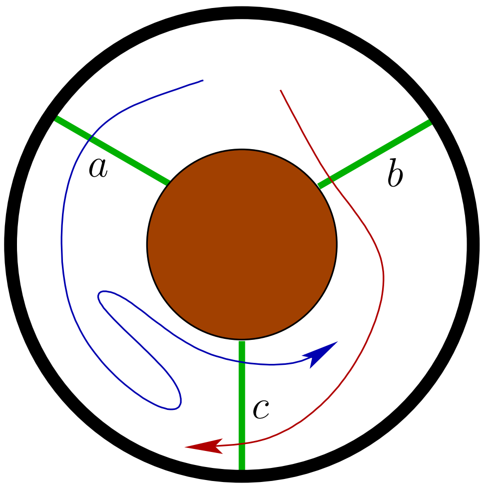
From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

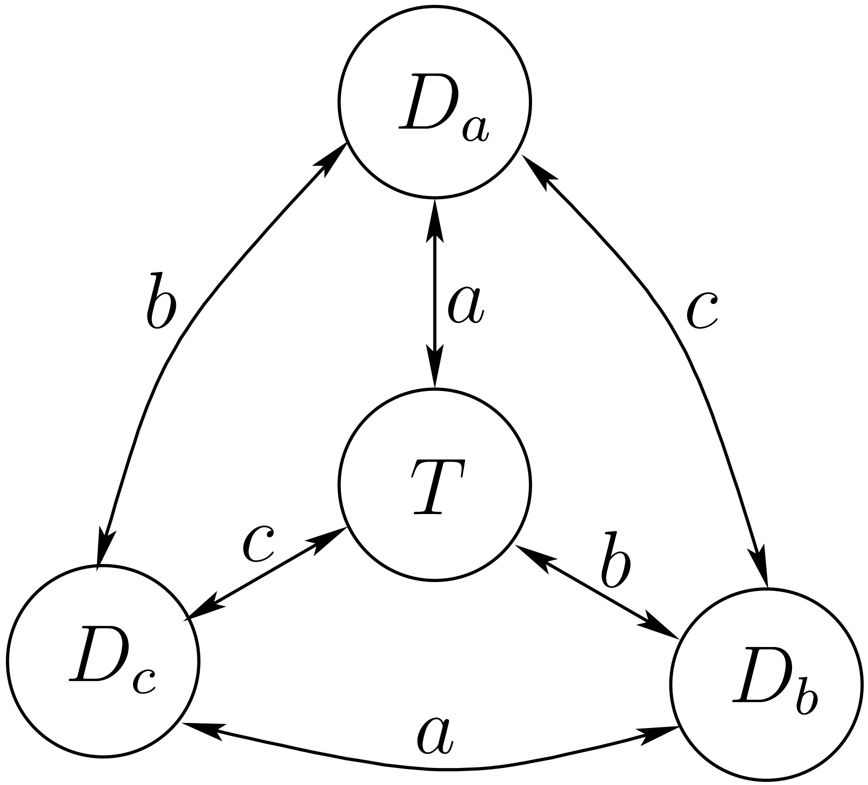
Controlling Wild Bodies

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Recall: Two-Bit Filter

Recall the simple filter that determines whether two bodies are in the same region.

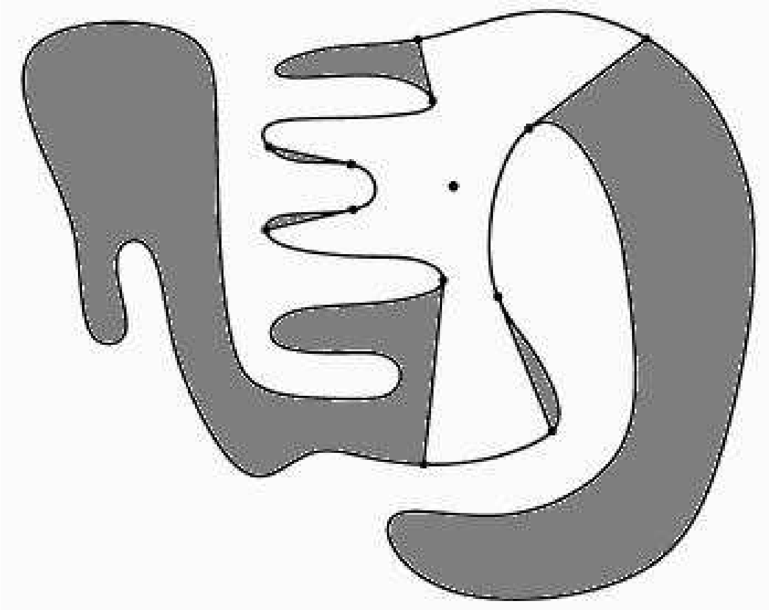


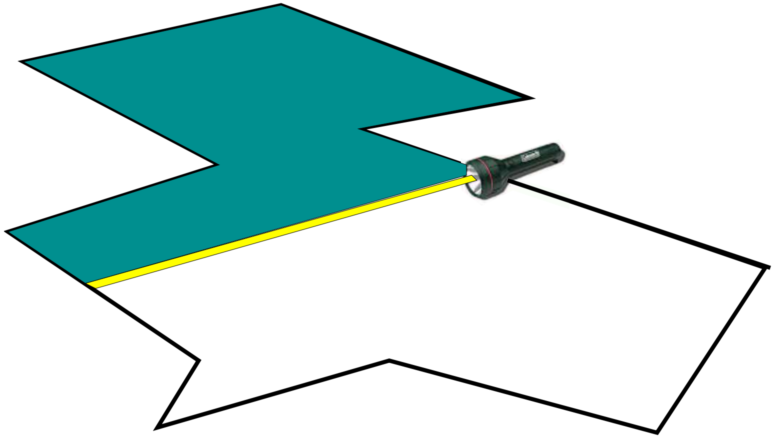


Recall: Visibility-Based Pursuit-Evasion

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

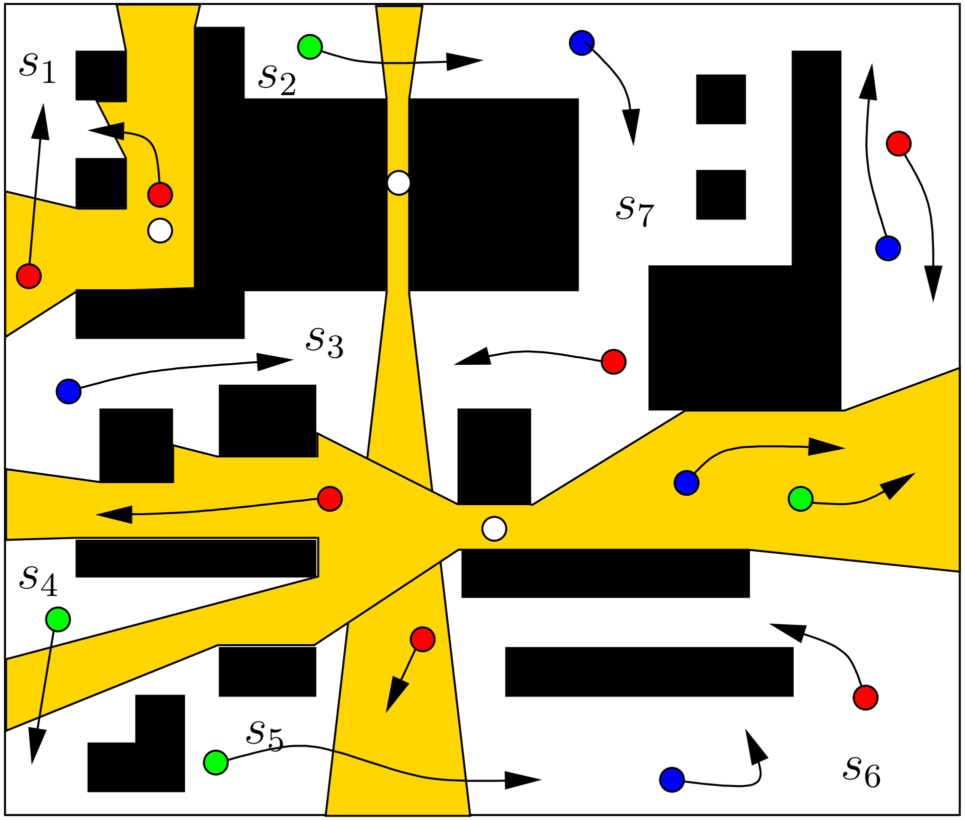
- A 2D environment, possibly curved
- Unpredictable point “evaders” move with unbounded speed
- Point “pursuers” use visibility sensors to find all evaders





Recall: Shadow Information Spaces

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies



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

How many are there? What kinds of bodies are there?

From Filtering to Actuation

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

- Key exploited property in filters: **Motion continuity**

From Filtering to Actuation

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

- Key exploited property in filters: **Motion continuity**
- Bring in **actuation**, but continue with minimalism, reduced I-spaces
- Passive $\longrightarrow$ Avoid state estimation

  Active $\longrightarrow$ **Avoid system identification**

- What is the new key property?

From Filtering to Actuation

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

- Key exploited property in filters: **Motion continuity**
- Bring in **actuation**, but continue with minimalism, reduced I-spaces
- Passive $\longrightarrow$ Avoid state estimation

  Active $\longrightarrow$ **Avoid system identification**

- What is the new key property? **Wildness**

Our Simple Robot

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Properties?

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

- No map is given in advance

Properties?

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

- No map is given in advance
- No position estimation is available

Properties?

- No map is given in advance
- No position estimation is available
- No system identification has been performed

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Properties?

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

- No map is given in advance
- No position estimation is available
- No system identification has been performed
- No sensors, inside or outside of the robot

Properties?

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

- No map is given in advance
- No position estimation is available
- No system identification has been performed
- No sensors, inside or outside of the robot
- No computer or any other digital devices

Properties?

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

- No map is given in advance
- No position estimation is available
- No system identification has been performed
- No sensors, inside or outside of the robot
- No computer or any other digital devices
- Only one motor, oscillating at 2Hz

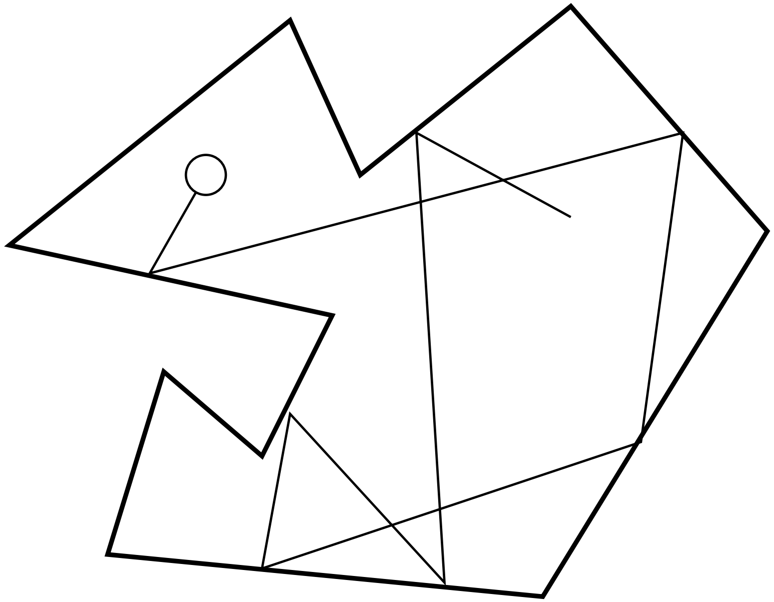
A Weasel Ball

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

An old, popular toy (costs about $4)

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

We say that a body is *wild* in a region $R \subseteq \mathbb{R}^2$ if it moves on a trajectory that causes it to repeatedly strike every open interval in ∂R (the boundary of R), with non-zero, non-tangential velocities.



Somewhat informal

Interesting, But How to Control?

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

- It is far from stable
- Almost impossible to predict
- Dynamical system modeling or identification is difficult

Interesting, But How to Control?

From filters to planning

General issues

Visibility-based pursuit evasion

Maze searching

Gap navigation trees

Learning convex hulls of landmarks

Bug algorithms

Sensorless manipulation

Controlling Wild Bodies

- It is far from stable
- Almost impossible to predict
- Dynamical system modeling or identification is difficult

Hmm...the situation is similar for humans.

Manipulating Humans with Gentle Guidance

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

How to clear out the breakfast area after 9:30am?

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Other Examples

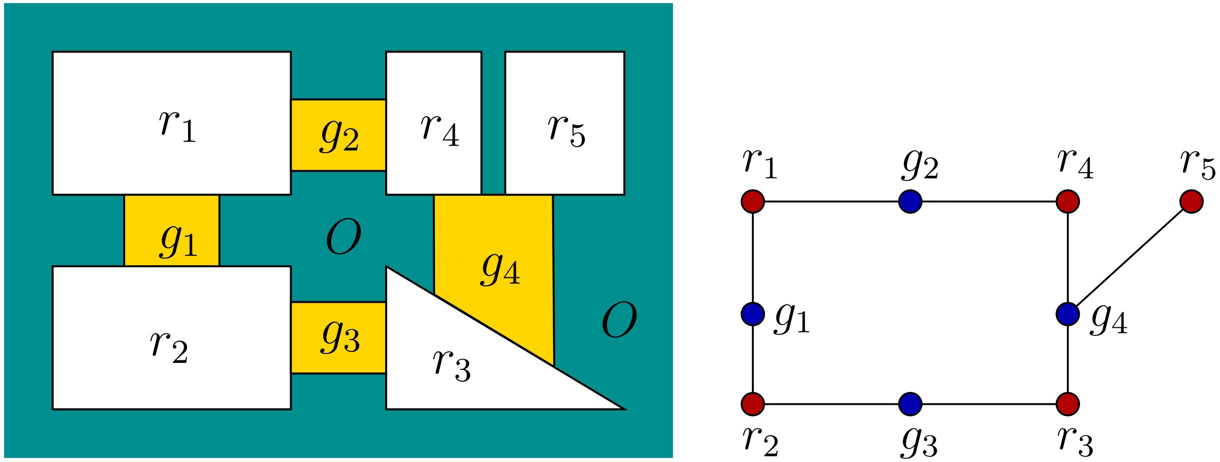
Also: bug traps

Some Related Research

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

- Tray tilting, Mason, Erdmann, 1988
- Virtual fences for herding cows, Butler, Corke, Peterson, Rus, 2004.
- Manipulation by vibration, Canny, Reznick, 1998; Vose, Lynch, 2011
- Building evacuation, Chalmet, Francis, Saunders, *Fire Technology*, 1982.

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies



The plane $\mathbb{R}^2$ is partitioned into:
1) *obstacle* set, 2) finite set of *regions*, 3) finite set of *gates*.

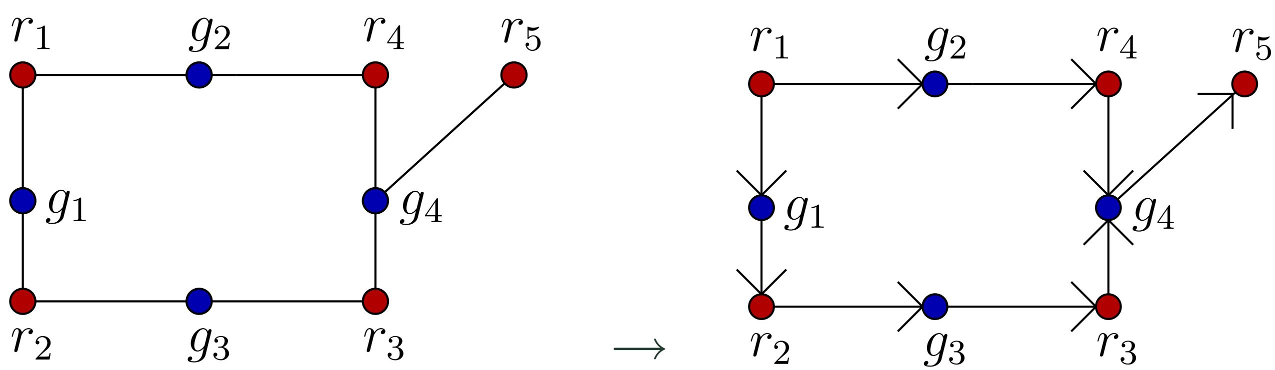
A bipartite graph represents the connectivity.

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

- Design some “wild” bodies
- Place bodies into regions
- Design gates to control them at the region level

Imagine an unusual hybrid system

A discrete flow across regions can be obtained



Types of Gates

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

- **Static gates:** The gates are fixed in advance and allow one-way motions from region to region.

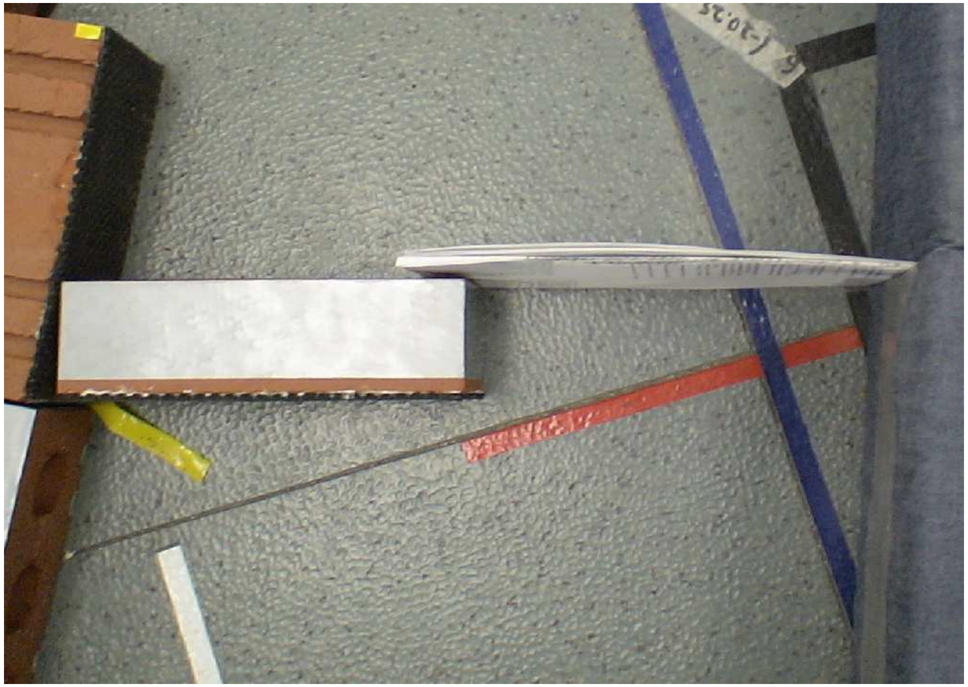
- **Pliant gates:** The gates have internal modes that affect how bodies are permitted to transition between regions and the modes may passively change via contact with bodies.

- **Controllable gates:** Based on information states, the gate modes are externally changed during execution.

- **Virtual gates:** Based on robot sensing, and never represent true physical obstructions.

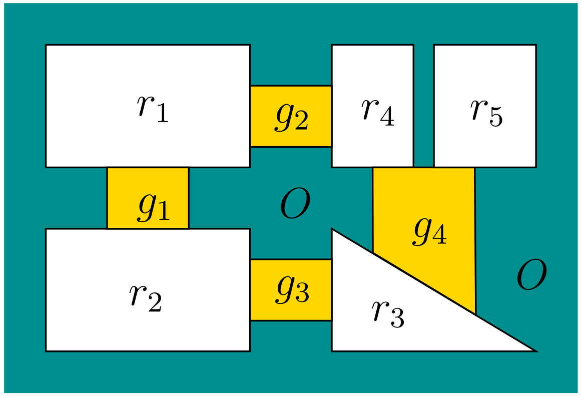
Engineering a Static Gate

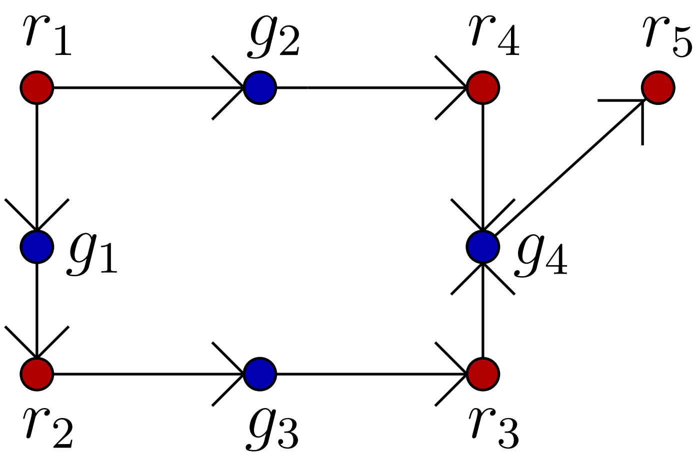
From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies



Strips of paper, wedged between bricks

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

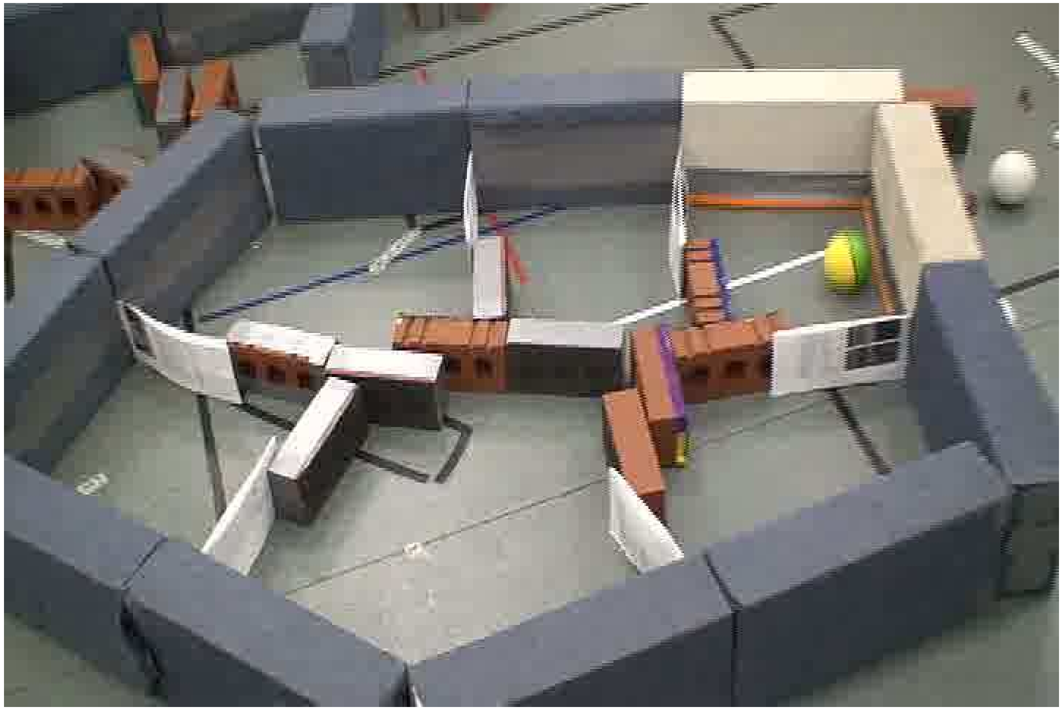




Compute a discrete flow to the goal region (BFS, Dijkstra).

Related work: Sequential composition of funnels, Lozano-Perez, Mason, Taylor, 1984; Mason, Goldberg, 1990; Burridge, Rizzi, Koditschek, 1999; Conner, Rizzi, Choset, 2003.

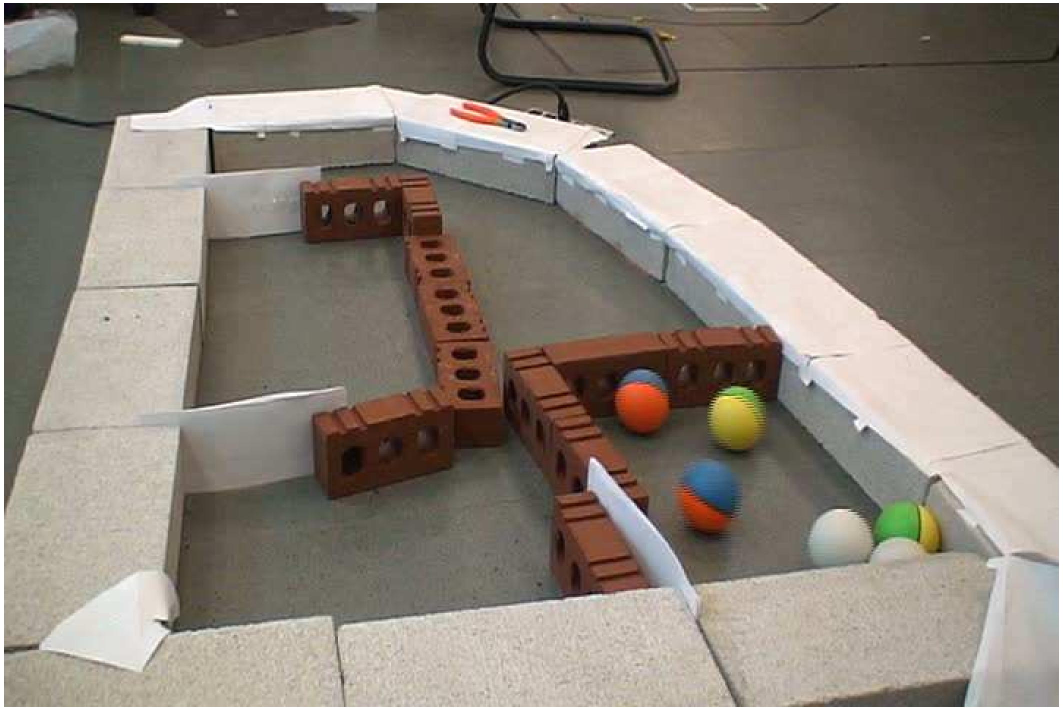
From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies



Goal: Flow to lower left region.

Static Gates: Multi-Body Navigation

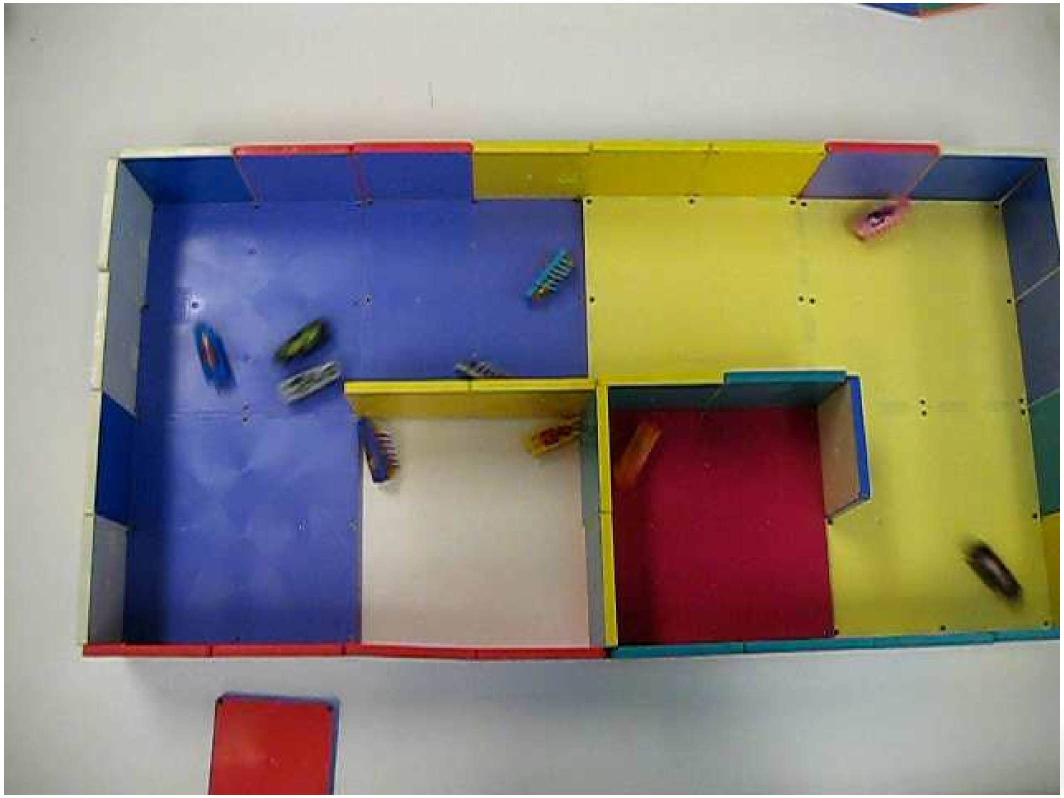
From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies



Six balls must flow to the upper right region.

Static Gates: Navigation with Hexbug Nanos

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies



Controlling 10 Hexbug Nanos.

Static Gates: Single-Body Patrolling

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Repeatedly travel a route through all regions.

Static Gates: Multi-Body Patrolling

From filters to planning

General issues

Visibility-based pursuit evasion

Maze searching

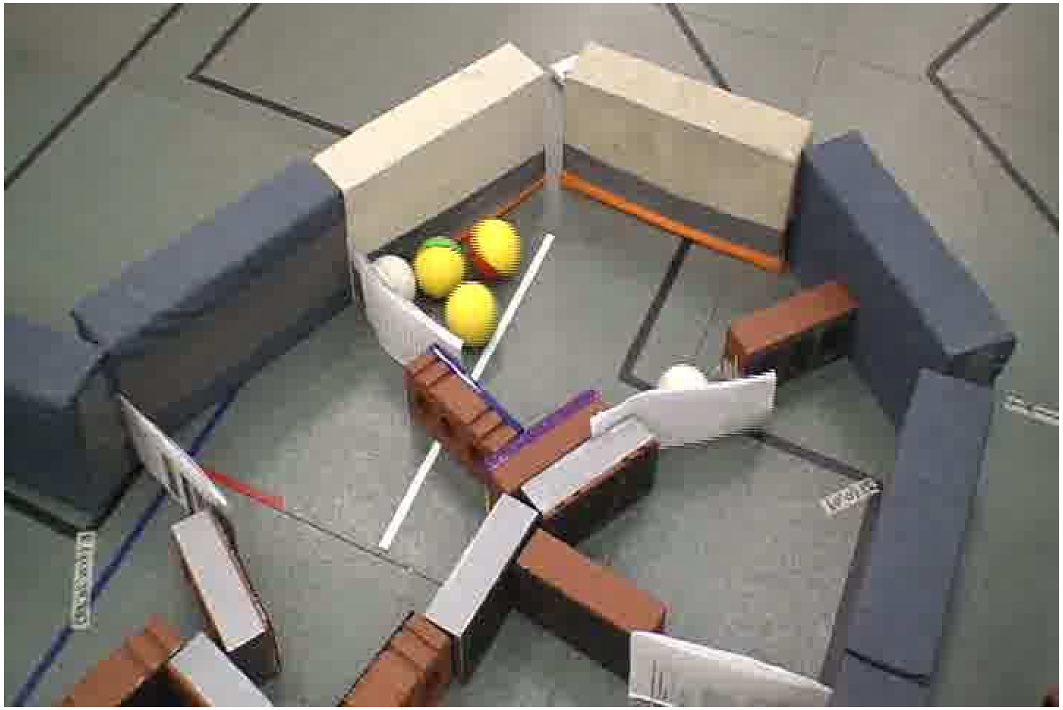
Gap navigation trees

Learning convex hulls of landmarks

Bug algorithms

Sensorless manipulation

Controlling Wild Bodies



Send all bodies on patrol, asynchronously.

50 Balls

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

A tale of 50 weaselballs...

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Pliant Gates

A pliant gate g has a finite set $M(g)$ of *modes*.

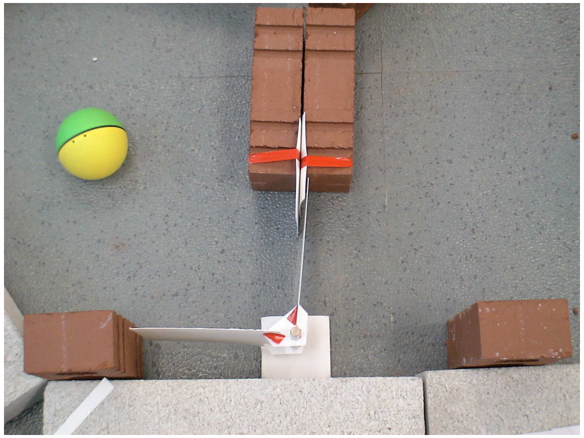
A body coming from region r into a gate g in mode m induces a *mode transition*:

$$m' = f(m, r)$$

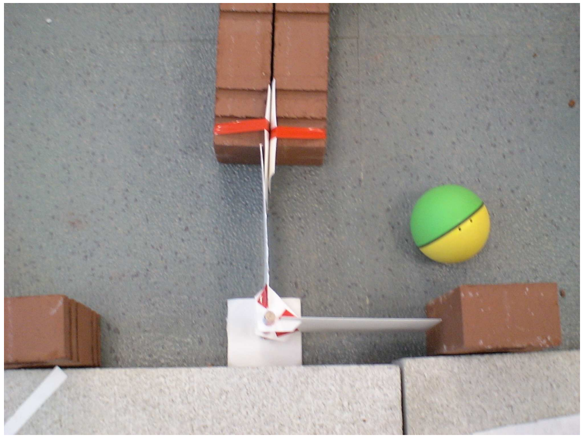
Mode transitions are caused by the bodies while they traverse gates.

A Simple Pliant Gate Design

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies



Left to right



Right to left

A two-mode pliant gate that maintains region counts.

Pliant Gates: Two-Way Revolving Door

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Keeping the number of balls roughly constant in each region.

Pliant Gates: Four-Way Revolving Door

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Controllable Gates

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Now suppose that the mode can be externally set by actuators.
Example modes $M(g)$ per gate g:

1. Block all passage
2. Allow left to right passage only
3. Allow right to left passage only
4. Allow bidirectional passage

Let M be the Cartesian product of all mode sets.

Key issue: What *information* is used to set $m \in M$?

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

For some time interval $T = [0, t]$:

$$\pi : T \rightarrow M$$

Time feedback

For sensor with observation space Y:

$$\pi : Y \rightarrow M$$

Sensor feedback

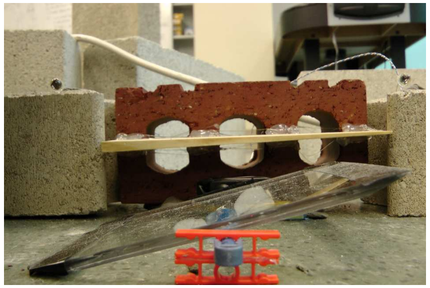
More generally, for any information space $\mathcal{I}$, we have:

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

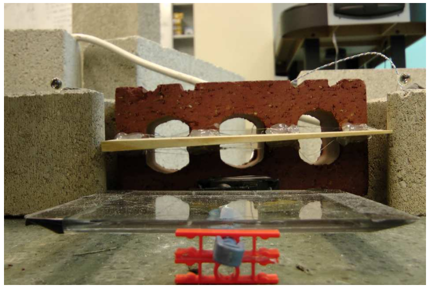
Sensor Feedback and Tilting Ramps

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

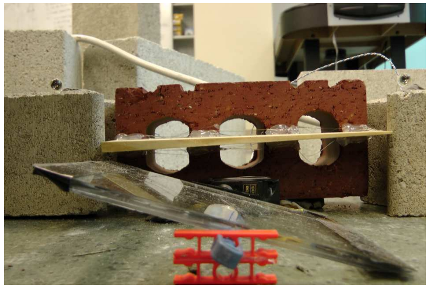
Tilting ramp:



L to R

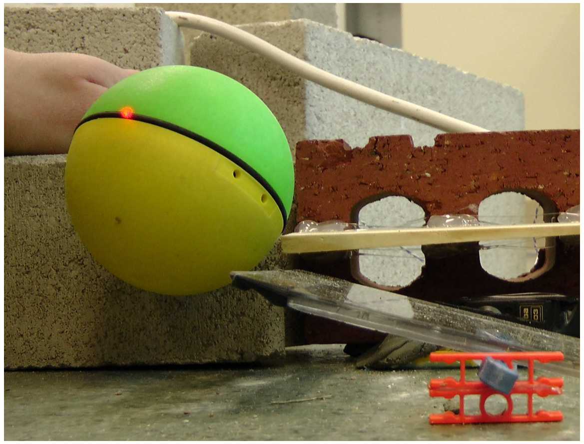


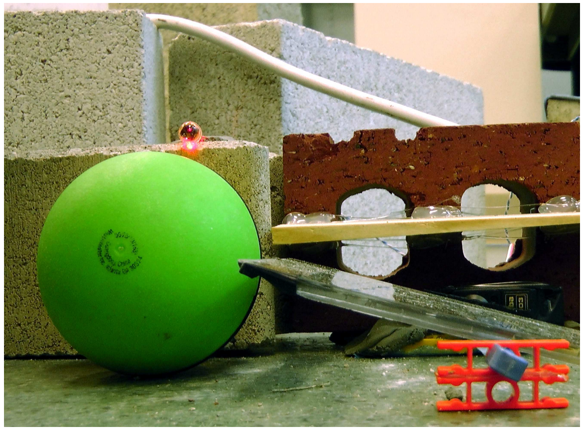
Blocked



R to L

Sensor beam feedback:





What Kinds of Tasks Can We Solve?

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Consider Linear Temporal Logic (LTL):

- Navigation: $\Diamond \pi_1$
- Sequencing: $\Diamond(\pi_1 \wedge \Diamond(\pi_2 \wedge \Diamond(\pi_3 \wedge \cdots \Diamond \pi_k) \cdots))$
- Coverage: $\Diamond \pi_1 \wedge \Diamond \pi_2 \wedge \cdots \Diamond \pi_k$
- Avoiding regions: $\neg(\pi_1 \vee \pi_2 \cdots \vee \pi_k) \mathcal{U} \pi_{final}$
- Patrolling: $\Box(\Diamond \pi_1 \wedge \Diamond \pi_2 \wedge \ldots \Diamond \pi_k)$.

Examples are from Kress-Gazit, Fainekos, Pappas, 2005.

RSS 2011: From LTL to weaselball implementations.

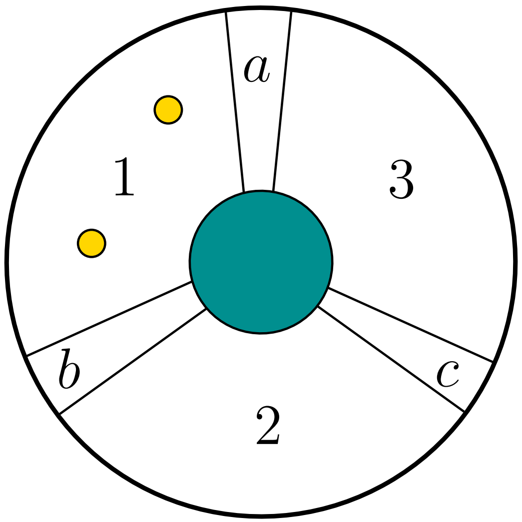
What Kinds of Tasks Can We Solve?

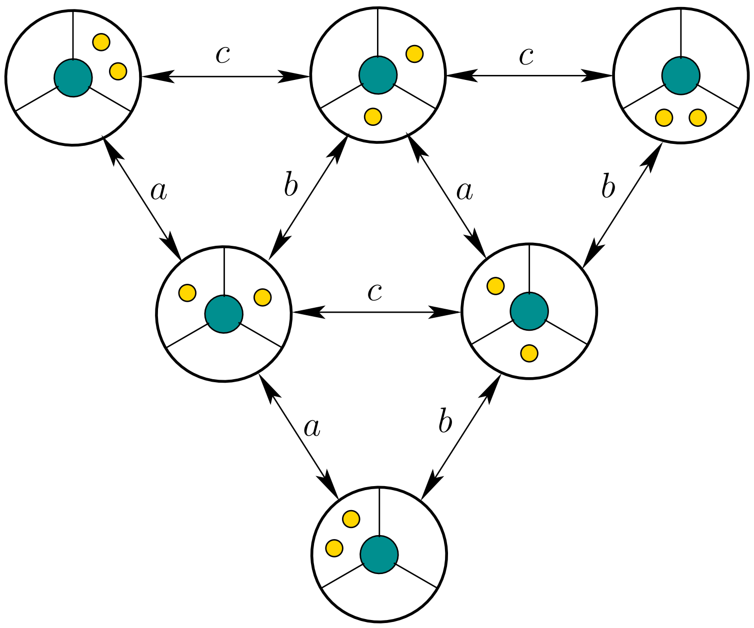
From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Approach:

1. Express the task in some logic
2. Convert into a solution in terms of region sequences
3. Implement using controllable gates and sensor feedback

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

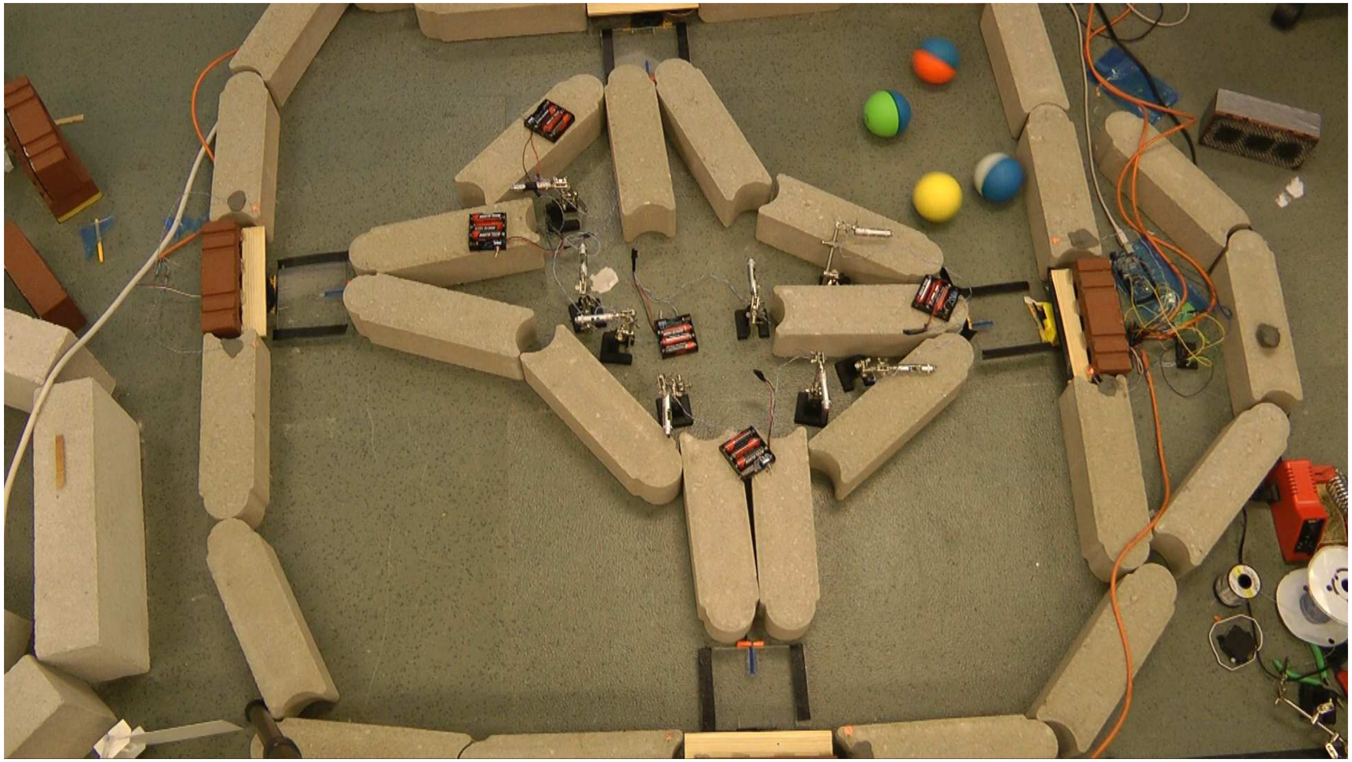




- Imagine indistinguishable balls in boxes.
- There is a natural transition graph.
- Express tasks using logic, and convert to sequences of distributions.

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

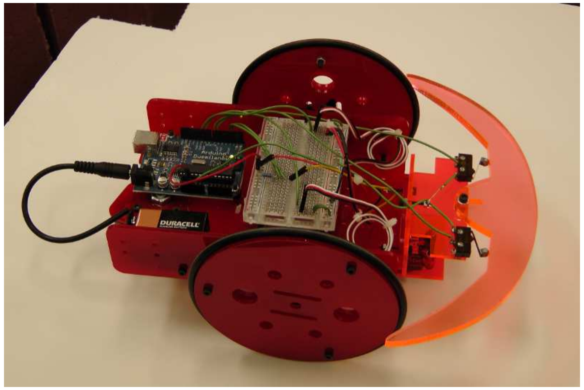
Splitting Video:



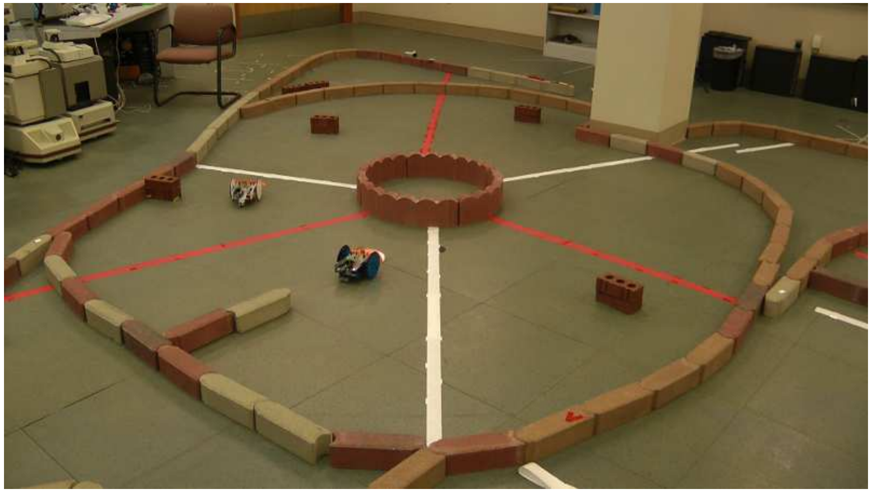
Merging Video

Virtual Gates

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies



A cheap color sensor detects a virtual gate crossing.



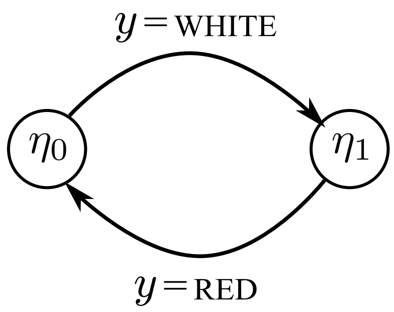
Communication allows simulation of a physical-gate system.

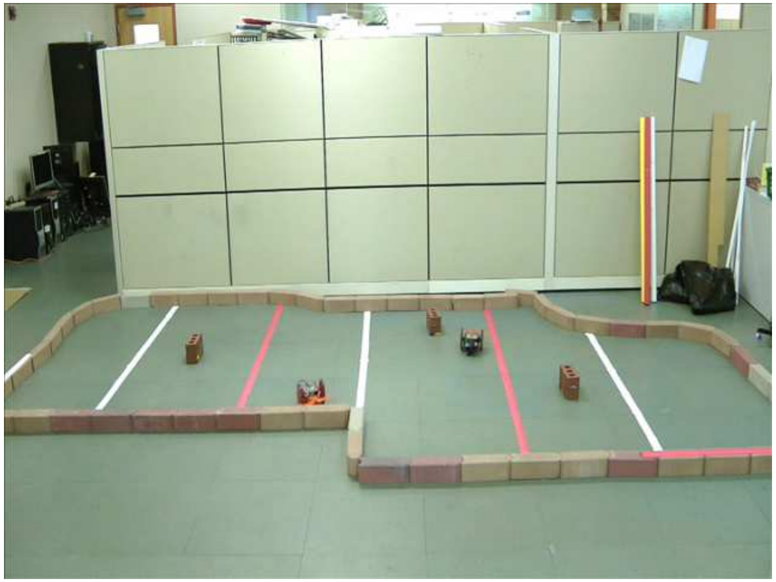
Virtual Gates

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

A simple information-feedback plan: $\pi : \mathcal{I} \rightarrow M$

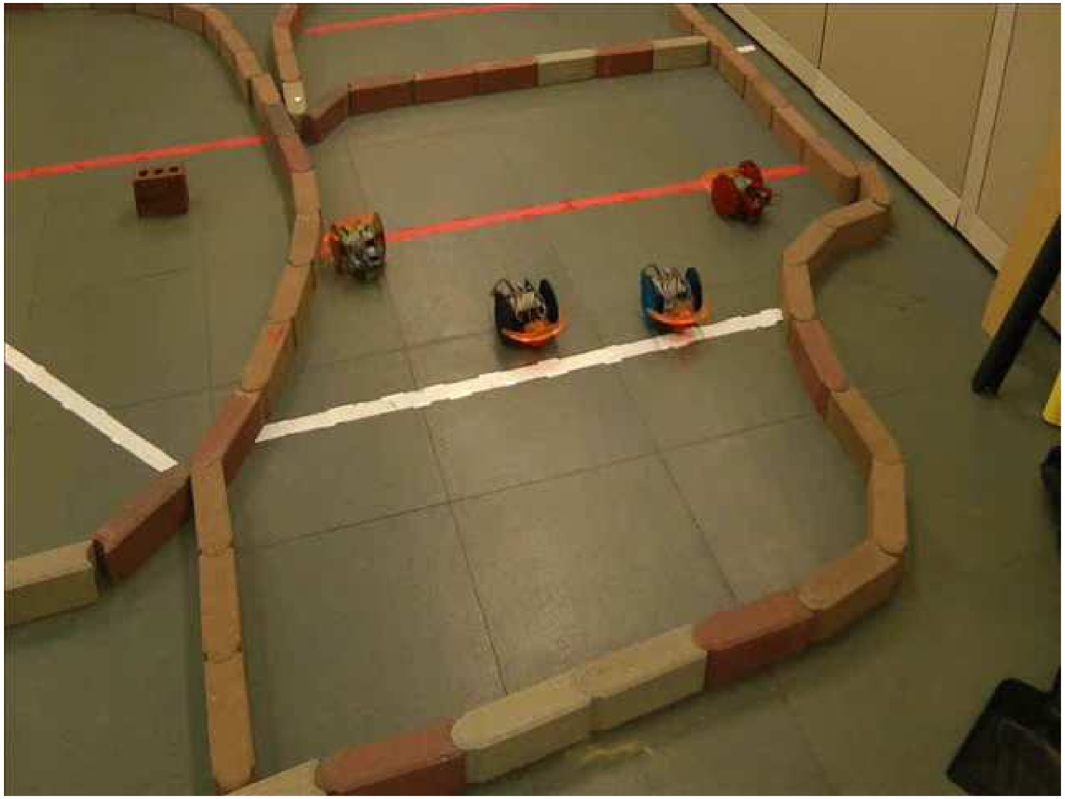
η_0: White open, red closed
η_1: White closed, red open





From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

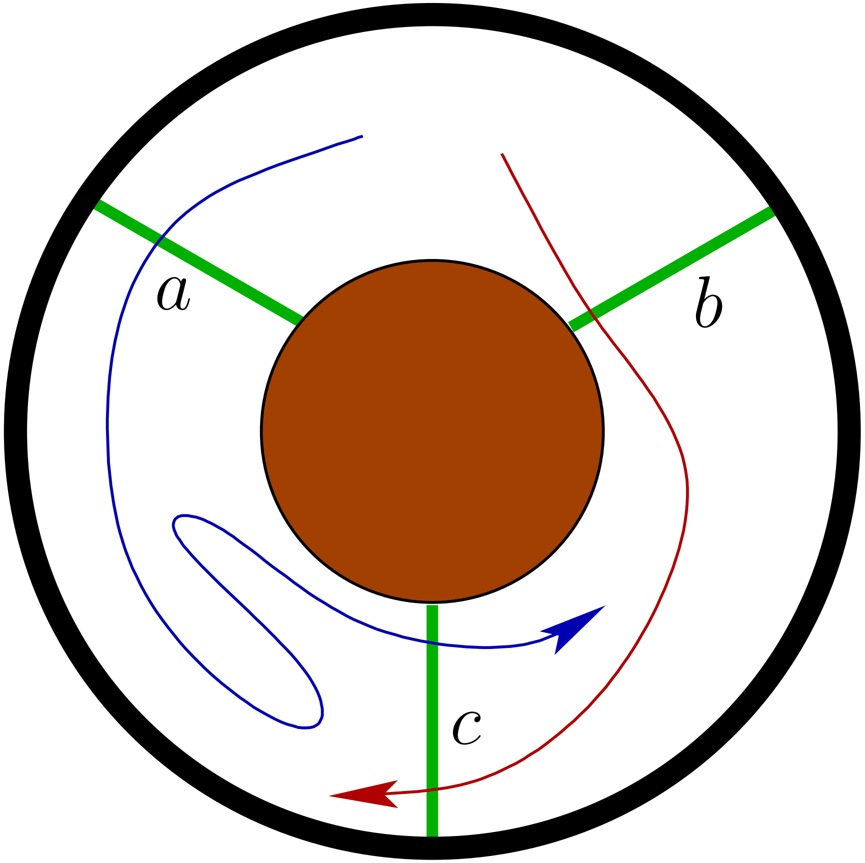
Separation into classes



Each robot can treat the boundaries (red and white) differently.

Information Feedback Using a Combinatorial Filter

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies



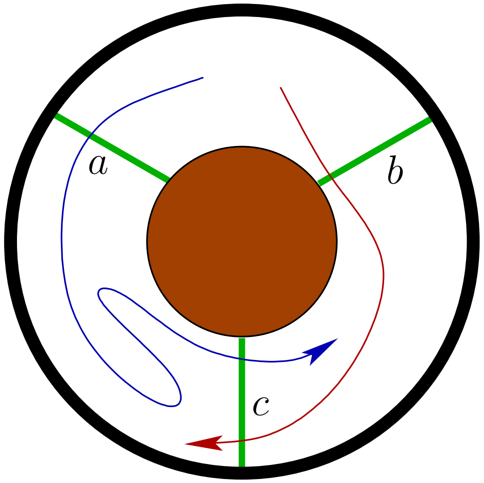
History I-state: $abbacbacabababcabcbba$

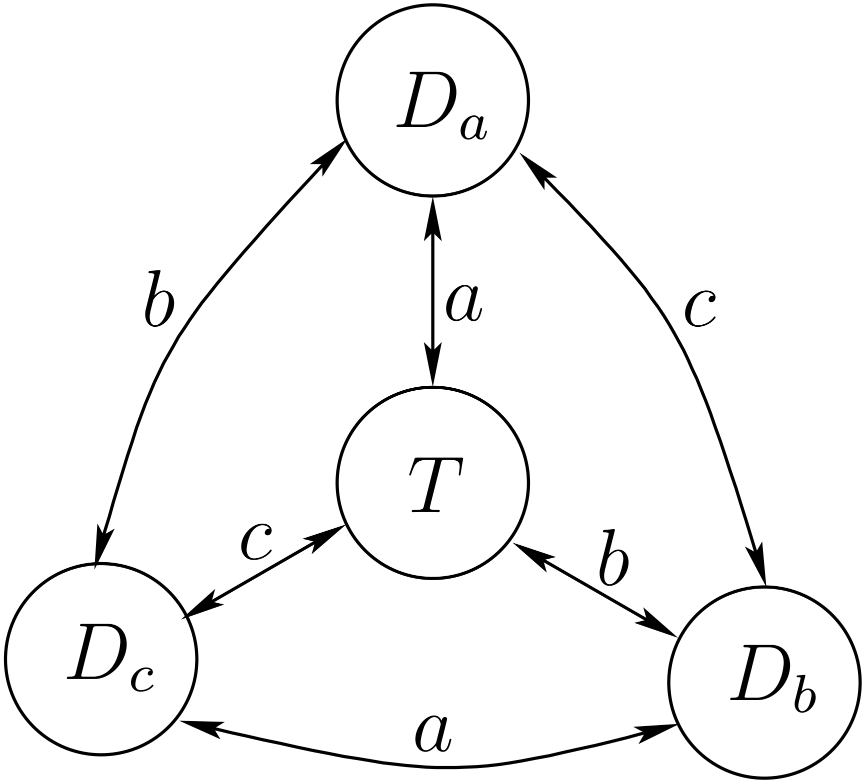
Question: Are the bodies in the **same** room?

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Living in a Tiny Information Space

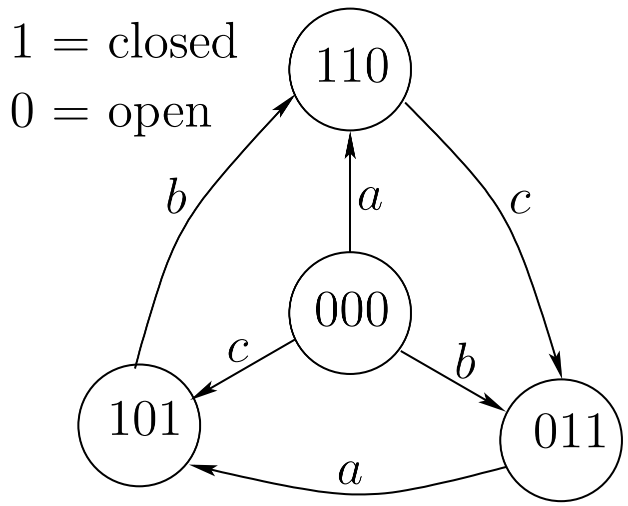
This two-bit machine can read strings of any length and correctly report the answer.

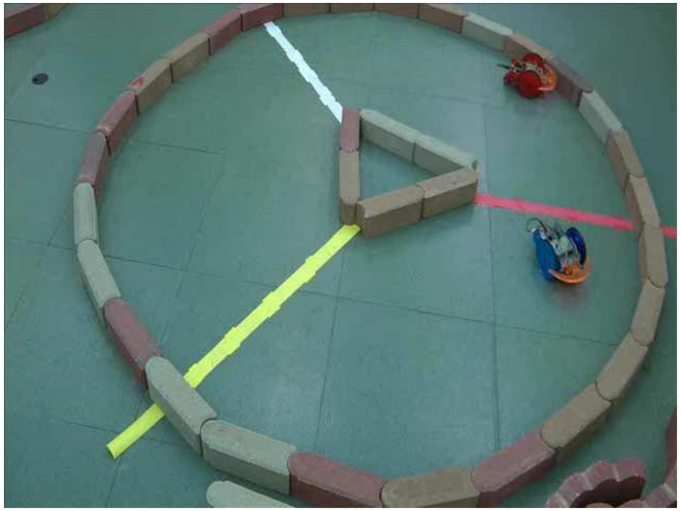




Three-Room Patrolling

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies





Information space: $\mathcal{I} = \{T, D_a, D_b, D_c\}$

Information feedback plan: $\pi : \mathcal{I} \to M$

Communication is needed between the robots.

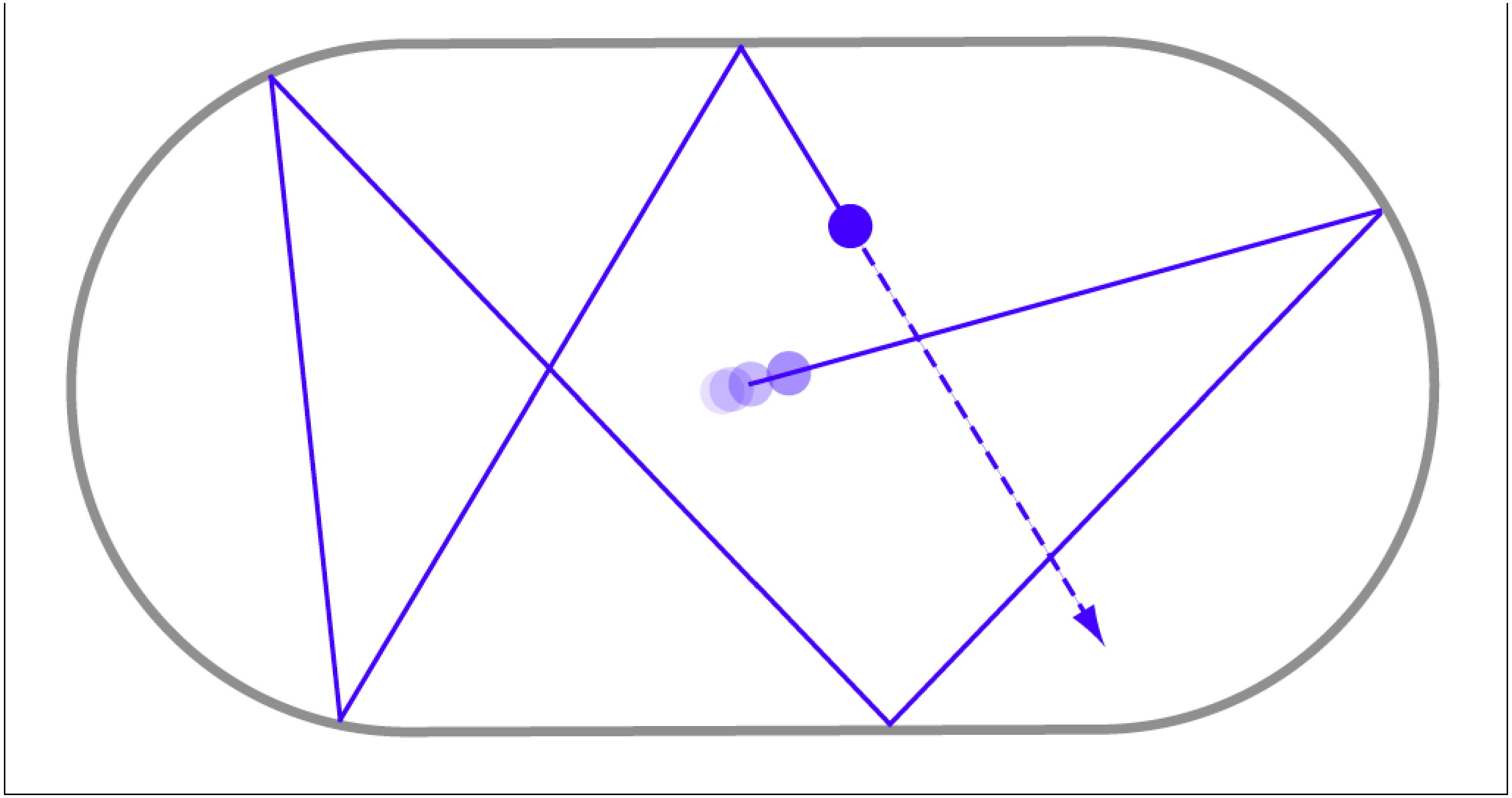
Theoretical Design and Analysis

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

- Connections to results in mathematics
- Performance analysis
- Designing better motions
- Optimal searching for the gate

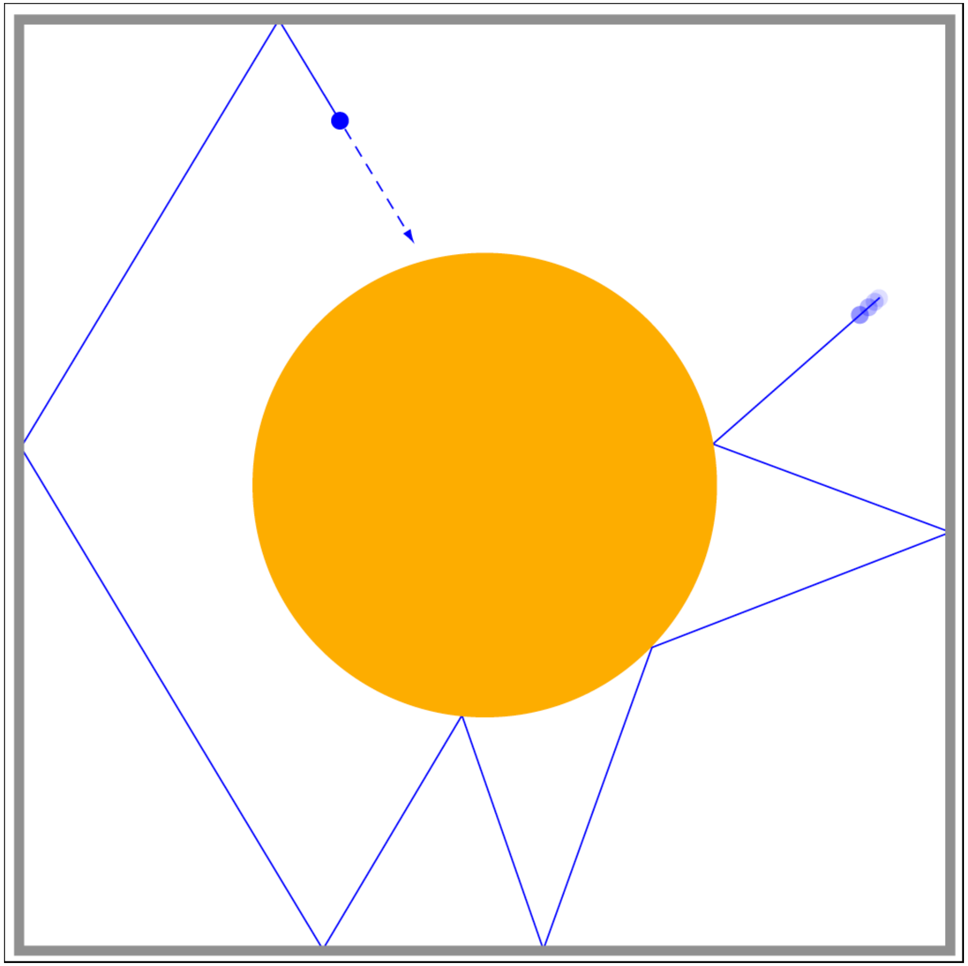
From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

History: Poincare, Hadamard, Artin, Sinai, Bunimovich, ...



Bunimovich stadium

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies



Sinai billiard

Ergodic Dynamics: Definition

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

First, a *measure-preserving dynamical system* is a four-tuple $(X, \mathcal{B}, \mu, T)$ for which: 1) X is a set, 2) $\mathcal{B}$ is a σ-algebra over X, 3) $\mu : \mathcal{B} \rightarrow [0, 1]$ is a measure, and 4) $T : X \rightarrow X$ is a measurable transformation that preserves measure (each $A \in \mathcal{B}$ satisfies $\mu(T^{-1}A) = \mu(A)$).

A measurable set $A \in \mathcal{B}$ is called T-invariant mod 0 if $\mu(T^{-1}(A) \bigtriangleup A) = 0$, in which $\bigtriangleup$ denotes the symmetric difference. Note that if this is true then A is T^n-invariant mod 0 for all n.

T is *ergodic* if for every T-invariant mod 0 measurable set A, we have $\mu(A) = 1$ or $\mu(A) = 0$.

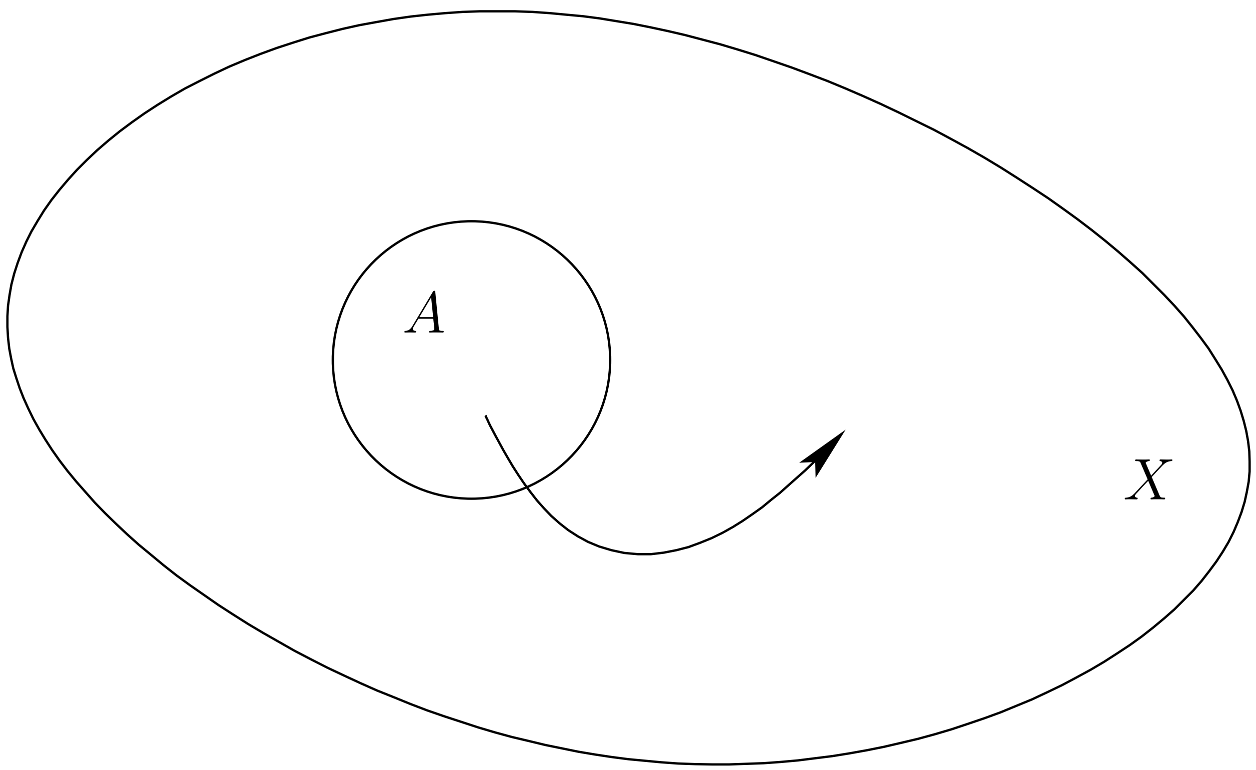
Intuition: You can't find a region (connected or not) that traps it.

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Ergodic Dynamics: Intuitive Definition

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

The system is a *measure-preserving transformation* $T : X \to X$ on a state space X.



You can't find a region A (connected or not) that traps the system, unless A or its complement has measure zero.

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

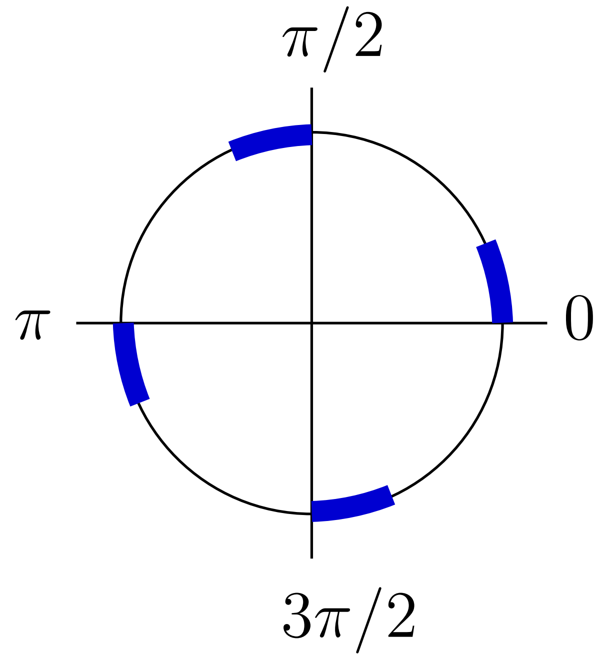
Ergodic Dynamics: Example

Let T be a planar rotation by angle θ.

$X = S^1$

If θ/π is irrational, then T is ergodic; otherwise, it is not.

Example: $\theta = \pi/2$



From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Let f by any μ-integrable function.

Time average:

$$\lim_{n \to \infty} \frac{1}{n} \sum_{k=0}^{n-1} f(T^k x)$$

Space average:

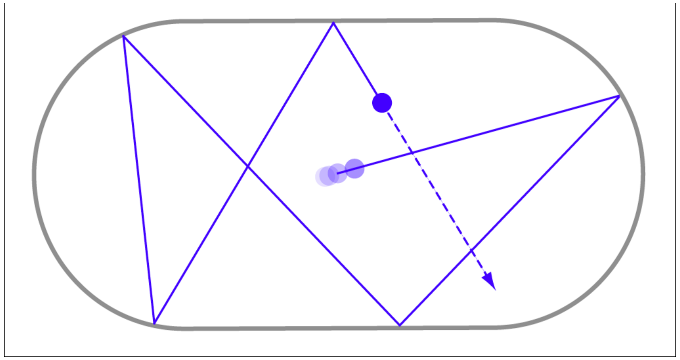
$$\frac{1}{\mu(X)} \int f d\mu$$

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Birkhoff (1931): If T is ergodic, then the time and space averages are the same (almost everywhere).

Example:
Take any $A \subseteq X$
Let $f(x) = 1$ if $x \in A$ and $f(1) = 0$ otherwise.

In this case, Birkoff's theorem states that the frequency of visits to A is equal to $\mu(A)$.



From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Ergodicity in Polygons

Kerckhoff, Masur, Smillie, 1986: For almost all polygons and almost all initial conditions, the billiard trajectory is ergodic.

What Is Different About Our Problems?

From filters to planning

General issues

Visibility-based pursuit evasion

Maze searching

Gap navigation trees

Learning convex hulls of landmarks

Bug algorithms

Sensorless manipulation

Controlling Wild Bodies

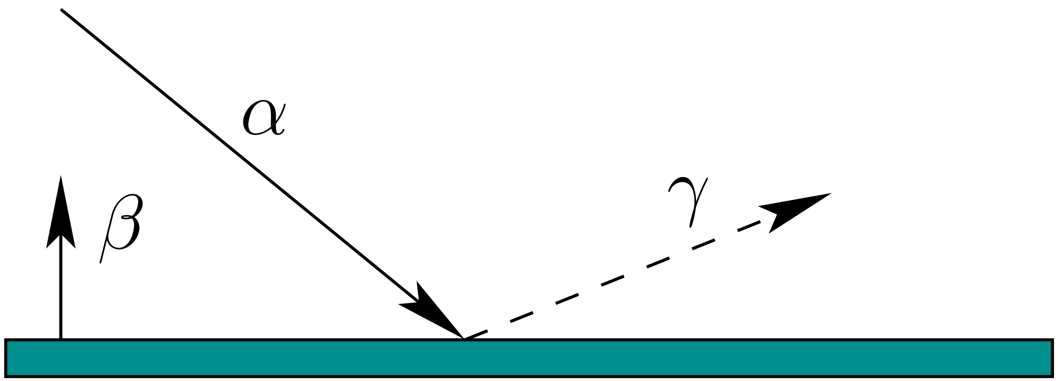
- We do not care about *measure-preserving* maps.
- There are many alternative ways to *bounce*.
- Classical ergodicity may be *overkill*.

This is ergodic almost everywhere, but not measure-preserving:

$$f : x \mapsto 2x \mod 1$$

Here, $f : [0,1] \rightarrow [0,1]$

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies



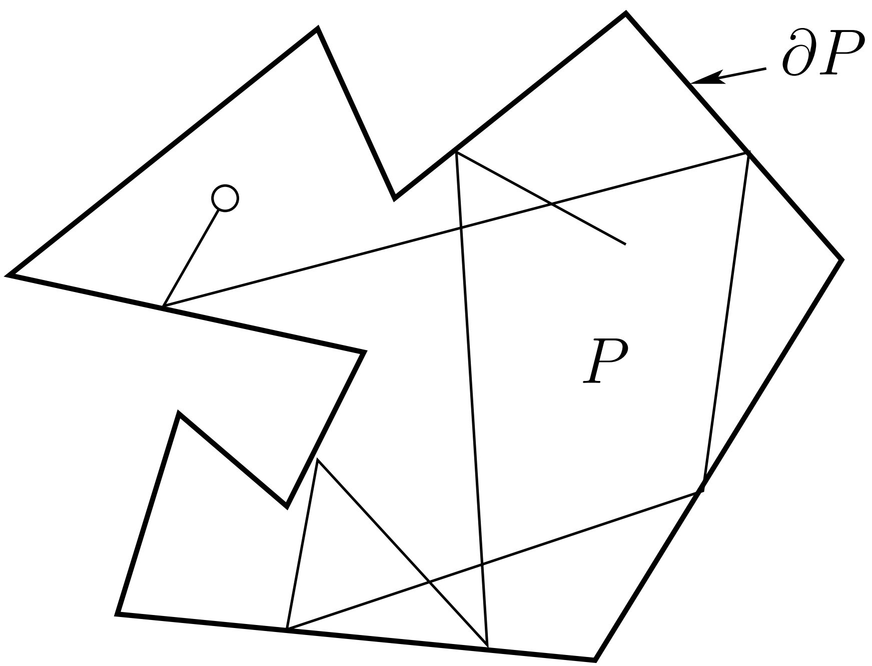
$\gamma = h(\alpha, \beta)$

Fundamental question: What sensors are needed?

Alternative: Select γ randomly (or with $p(\gamma|\alpha, \beta)$)

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Let $C \subseteq X$. Let $\tilde{x} : [0, \infty) \rightarrow X$ be a trajectory.
$\tilde{x}$ is called *topologically transitive with respect to* C if for every open set $O \subset C$, there exists a time $t > 0$ for which $\tilde{x}(t) \in O$.



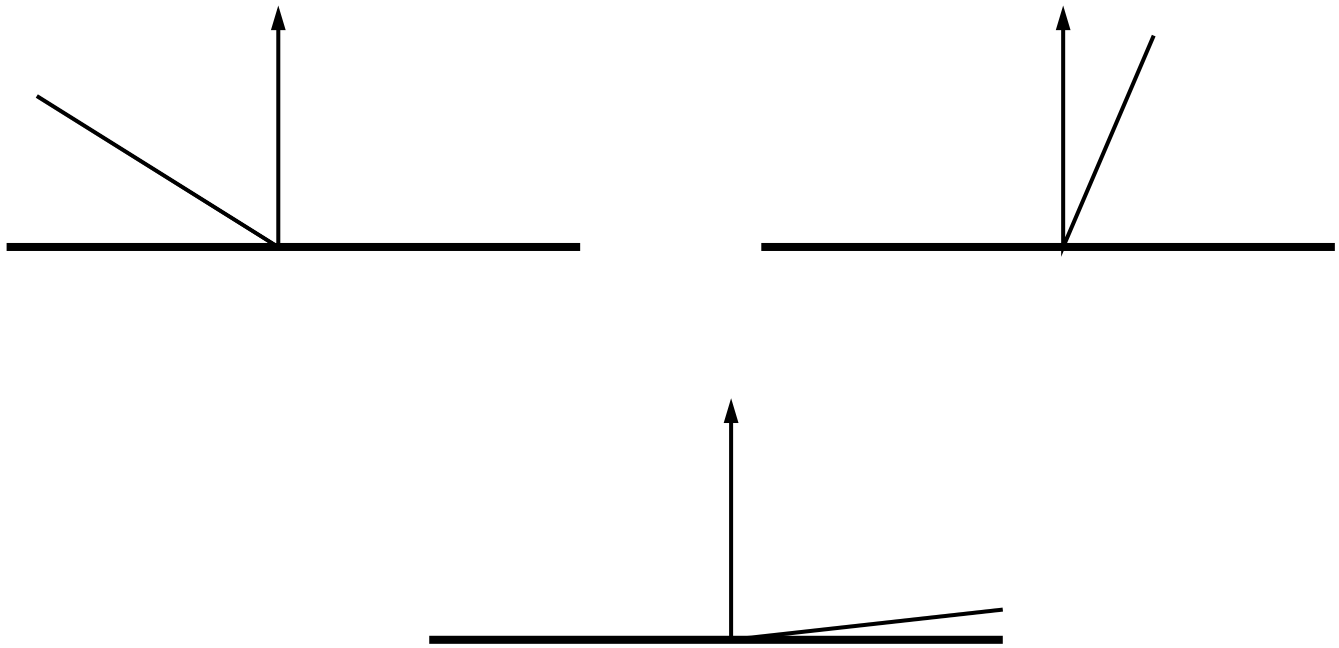
Suppose $X \subset \mathbb{R}^3$, in which $(x, y) \in P$ and $\theta \in S^1$.

Possibilities: $C = X$, $C = \partial P \times (0, \pi)$, or $C = \partial P$

Normal Bounce

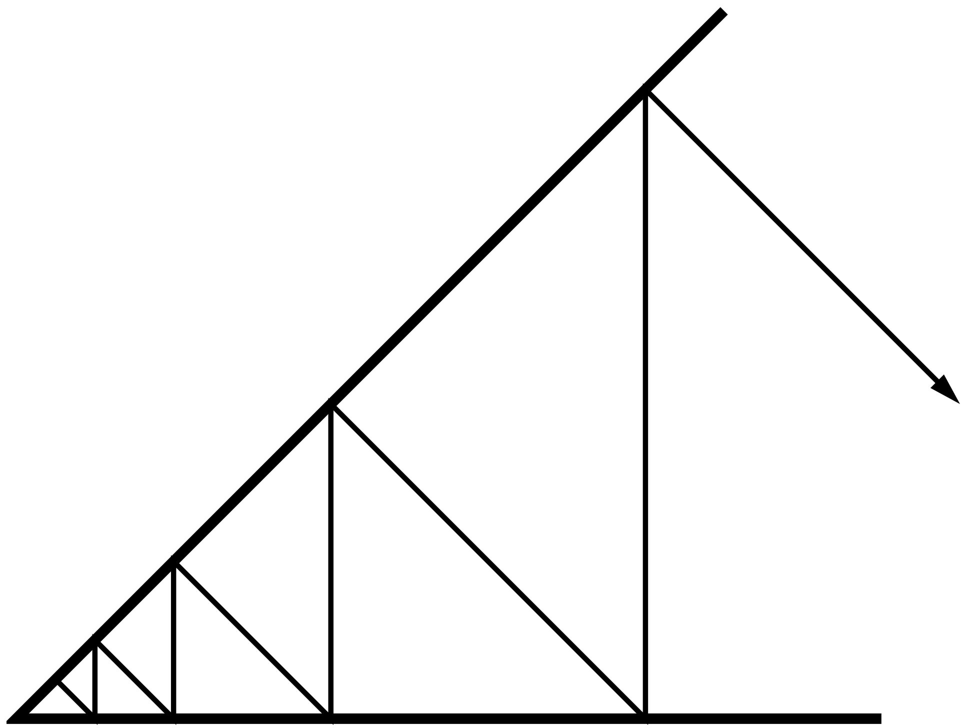
From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Angle of incoming body not relevant to trajectory after impact.



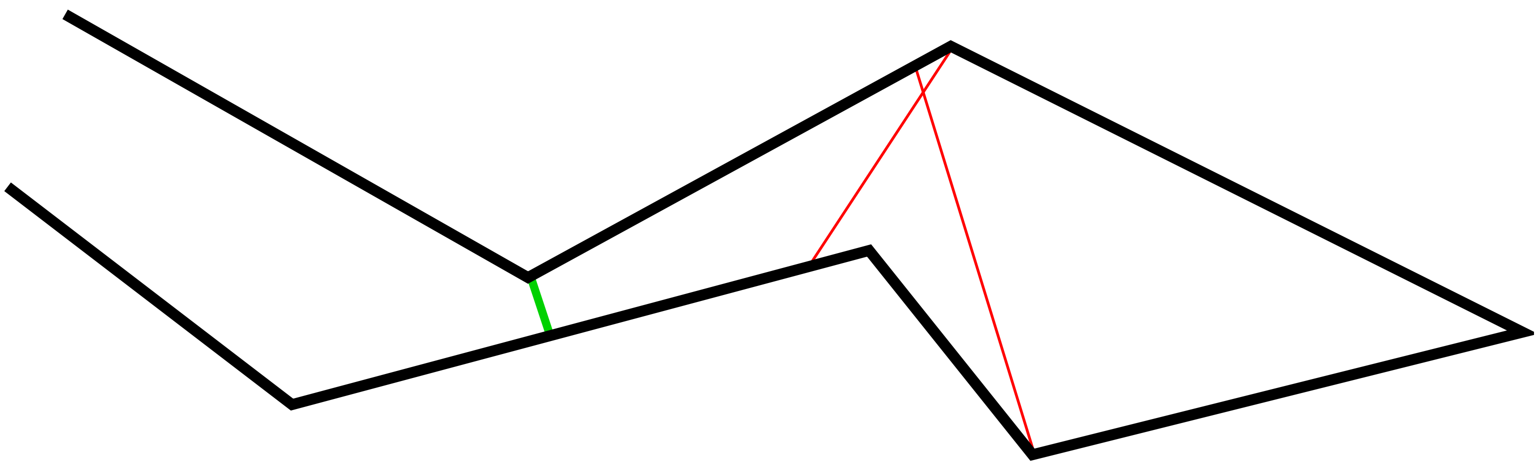
From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Bodies tend to move away from corners.



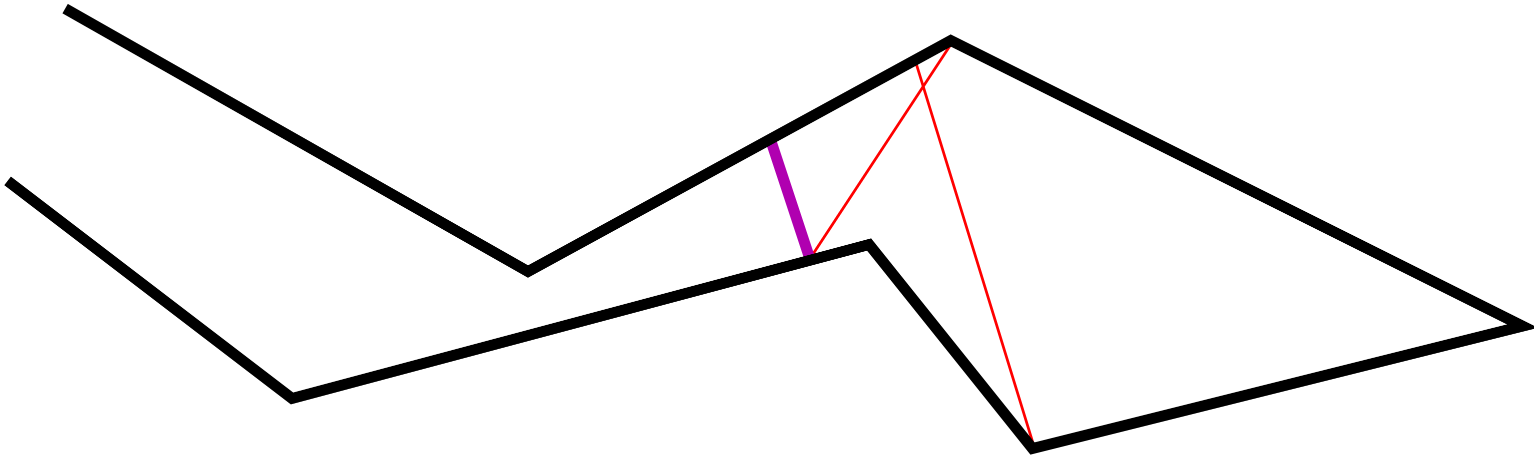
From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Nothing to the right of the green line will deflect a body back over the green line.



From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

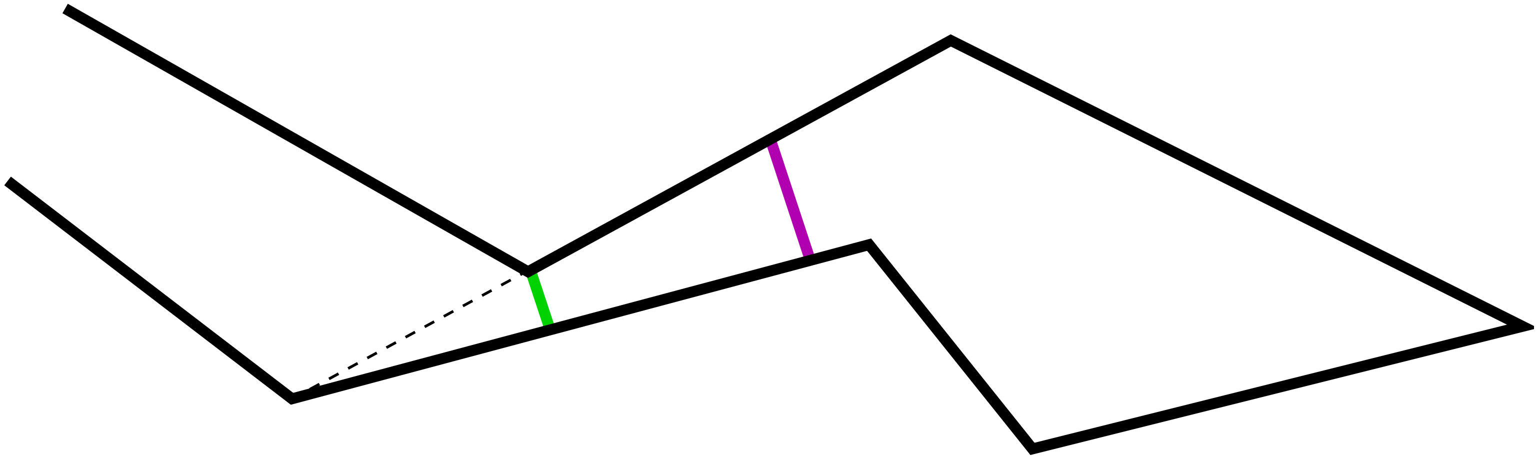
Nothing to the right of the purple line will deflect the body over the green line.



From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

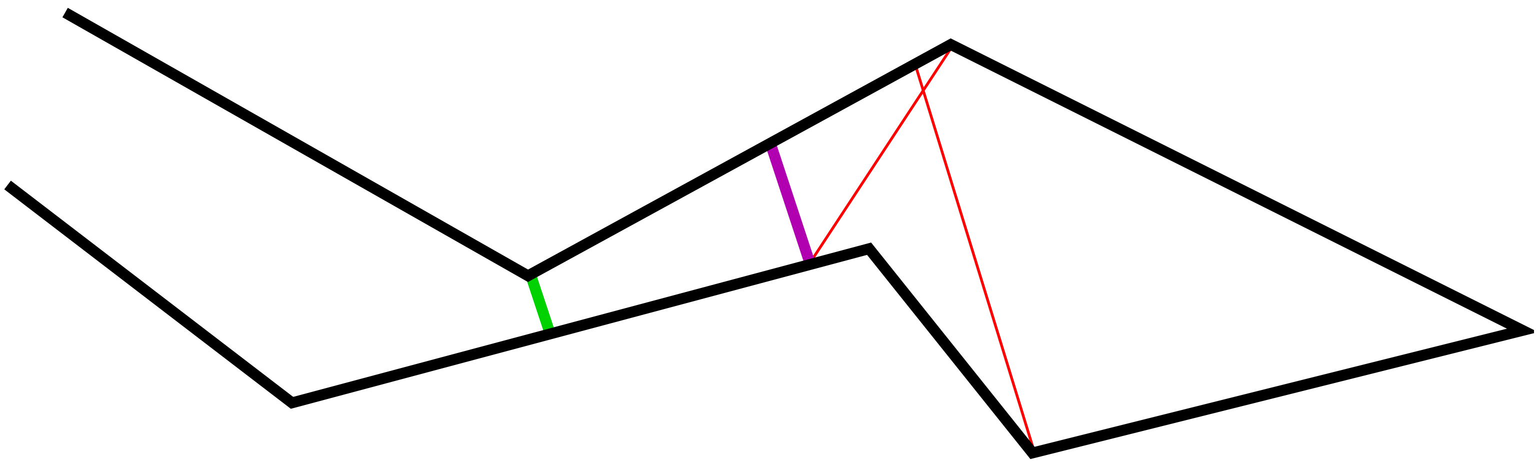
Trap

When the body crosses the green line, it moves toward the purple line, away from the “corner”.



From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

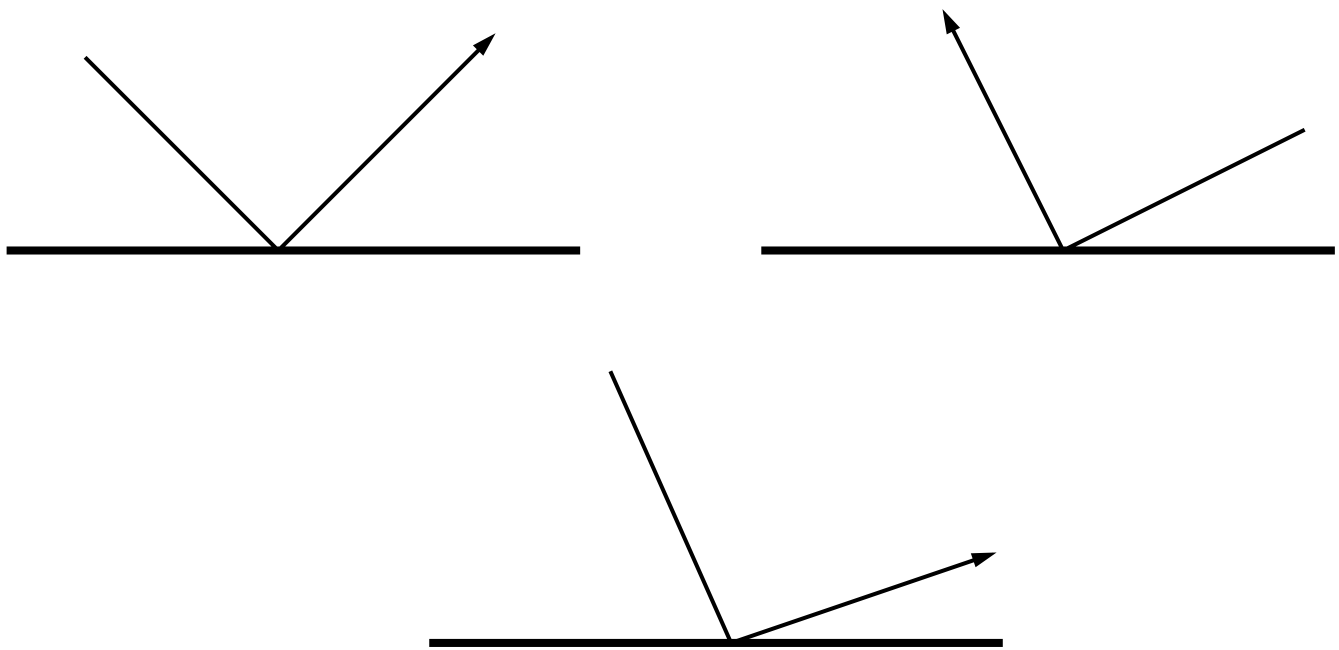
The green and purple lines denote the boundaries of a basin of attraction.



Right Angle Bounce

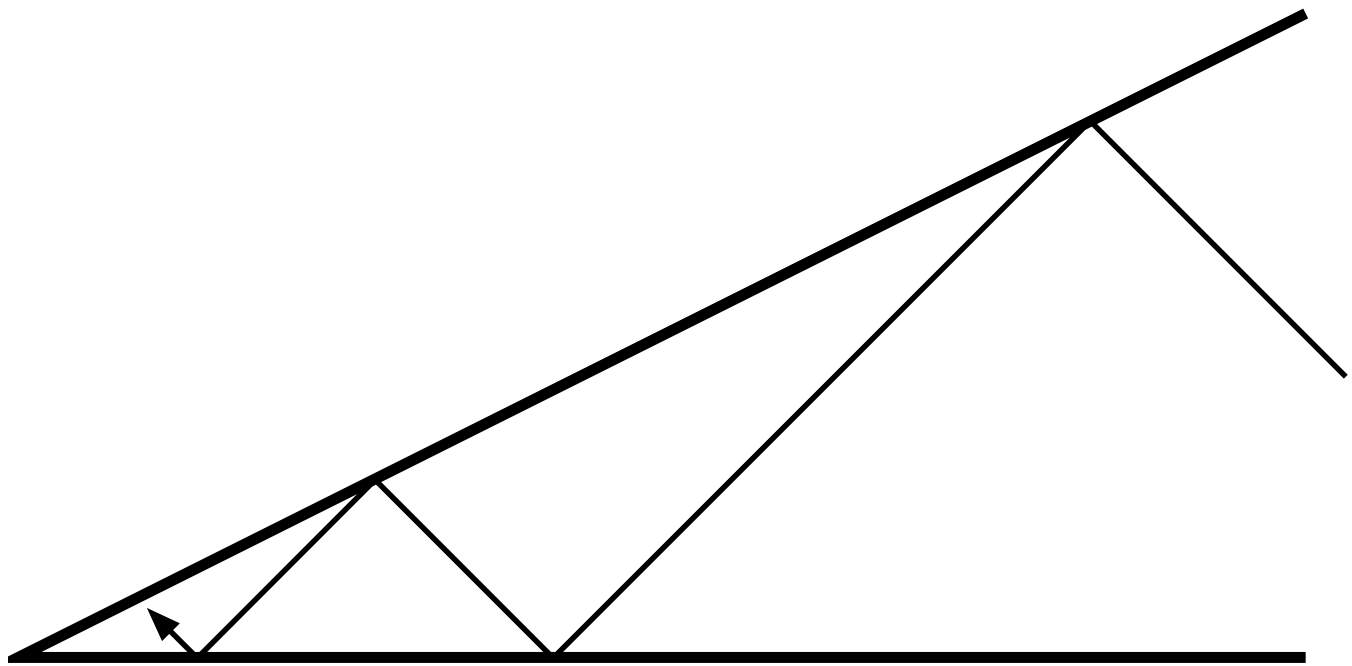
From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Body always reflects at a right angle, regardless of trajectory relative to wall.



From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

Right angle bouncing is attracted to corners.



Conclusions

From filters to planning
General issues
Visibility-based pursuit evasion
Maze searching
Gap navigation trees
Learning convex hulls of landmarks
Bug algorithms
Sensorless manipulation
Controlling Wild Bodies

The general paradigm:

- Let the bodies “run wild”, rather than stabilizing.
- Use physical or virtual gates to gently guide them.
- Use as little sensing and comminication as possible.

Challenges:

- Designing more systems of bodies and gates
- Characterizing the space of tasks that can be solved
- Development and analysis of simple bouncing primitives

Part 5 Summary

From filters to planning

General issues

Visibility-based pursuit evasion

Maze searching

Gap navigation trees

Learning convex hulls of landmarks

Bug algorithms

Sensorless manipulation

Controlling Wild Bodies

- Use filters to make I-space transitions
- Plan directly in the I-space
- General planning issues
- Need to design virtual sensors, filters, and planning around a task
- Several examples were shown

Although several examples of nice reduced-complexity I-spaces have been found, we have barely scratched the surface...