CS 4758/6758: Robot Learning

Spring 2010: Lecture 8

Ashutosh Saxena

Announcements

- HW3 posted. Due Mar 9 at 5pm.

- Project proposal
  - Everyone should have received feedback.
  - And should have access to the robot/lab by this Wednesday.

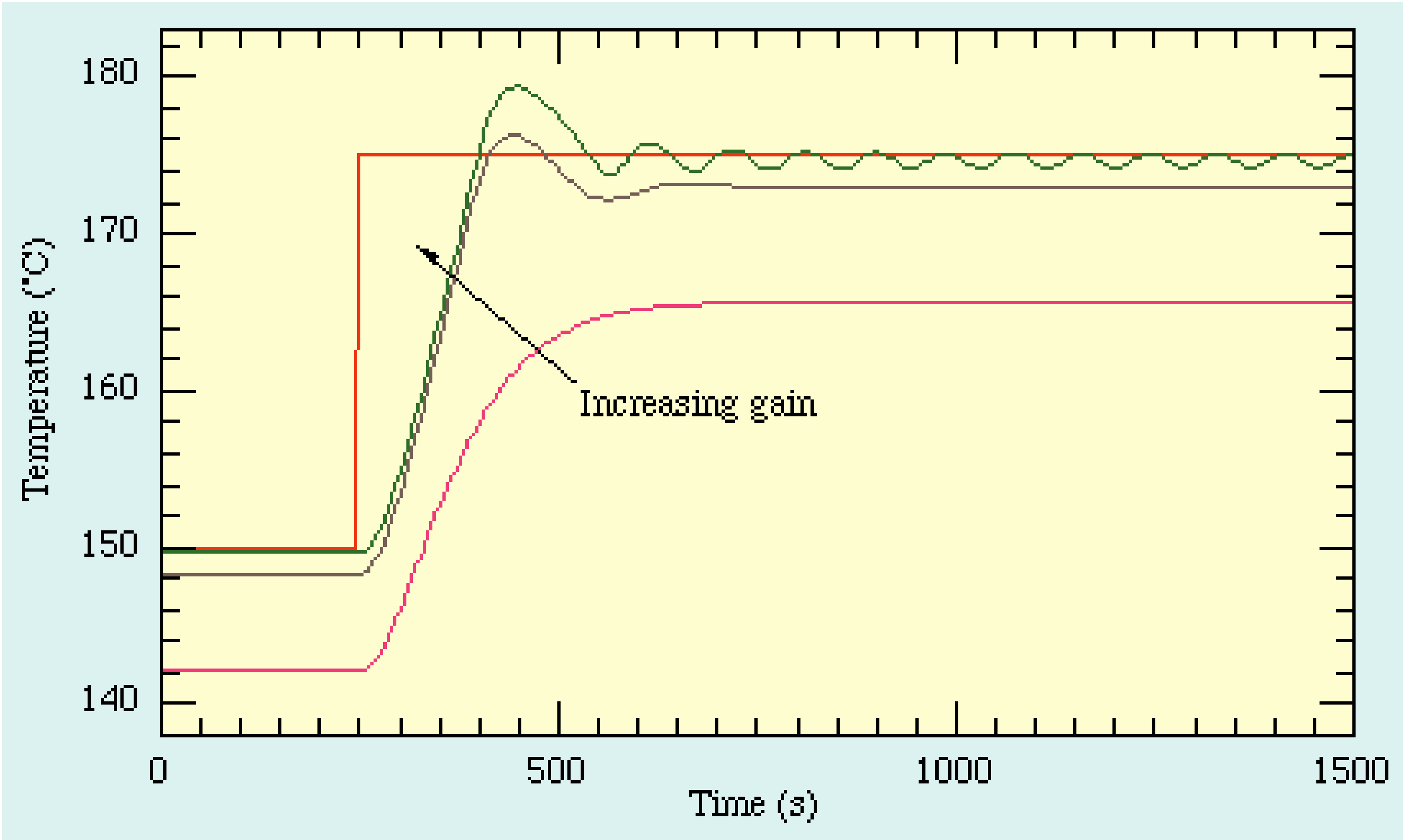
Proportional Control

- The setpoint x_{set} is the desired value.
  - The controller responds to error: $e = x - x_{set}$
- The goal is to set u to reach $e = 0$.

$$u = -ke$$

- Push back, *proportional* to the error.
- The controller gain k determines how quickly the system responds to error.

Proportional Control in Action



- Increasing gain approaches setpoint faster
- Can leads to overshoot, and even instability
- Steady-state offset

Integral Control

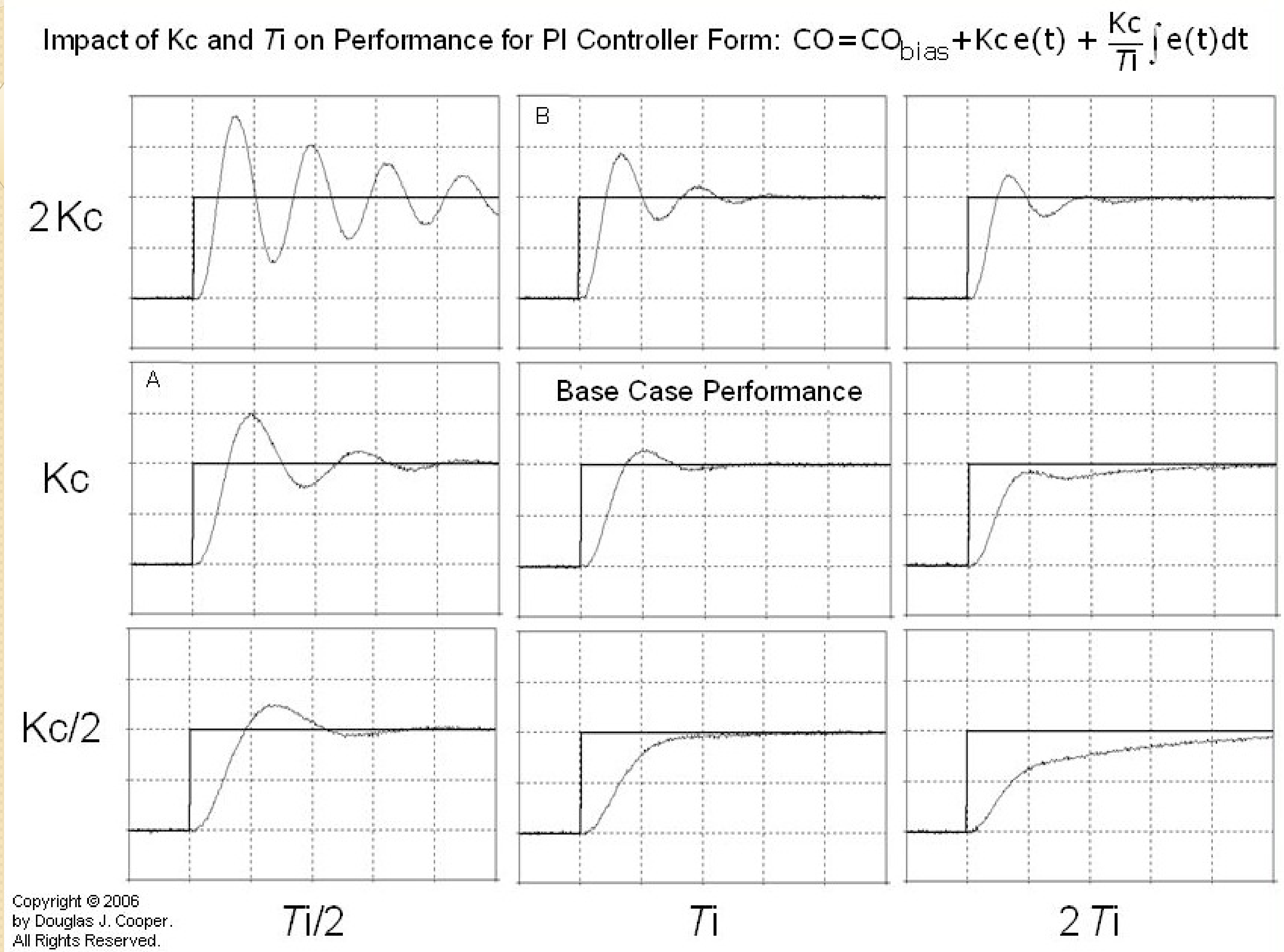
$$u_b(t) = -k_I \int_0^t e\,dt + u_b$$

- Therefore

$$u(t) = -k_P e(t) - k_I \int_0^t e\,dt + u_b$$

- The Proportional-Integral (PI) Controller.

Exploring PI Control Tuning



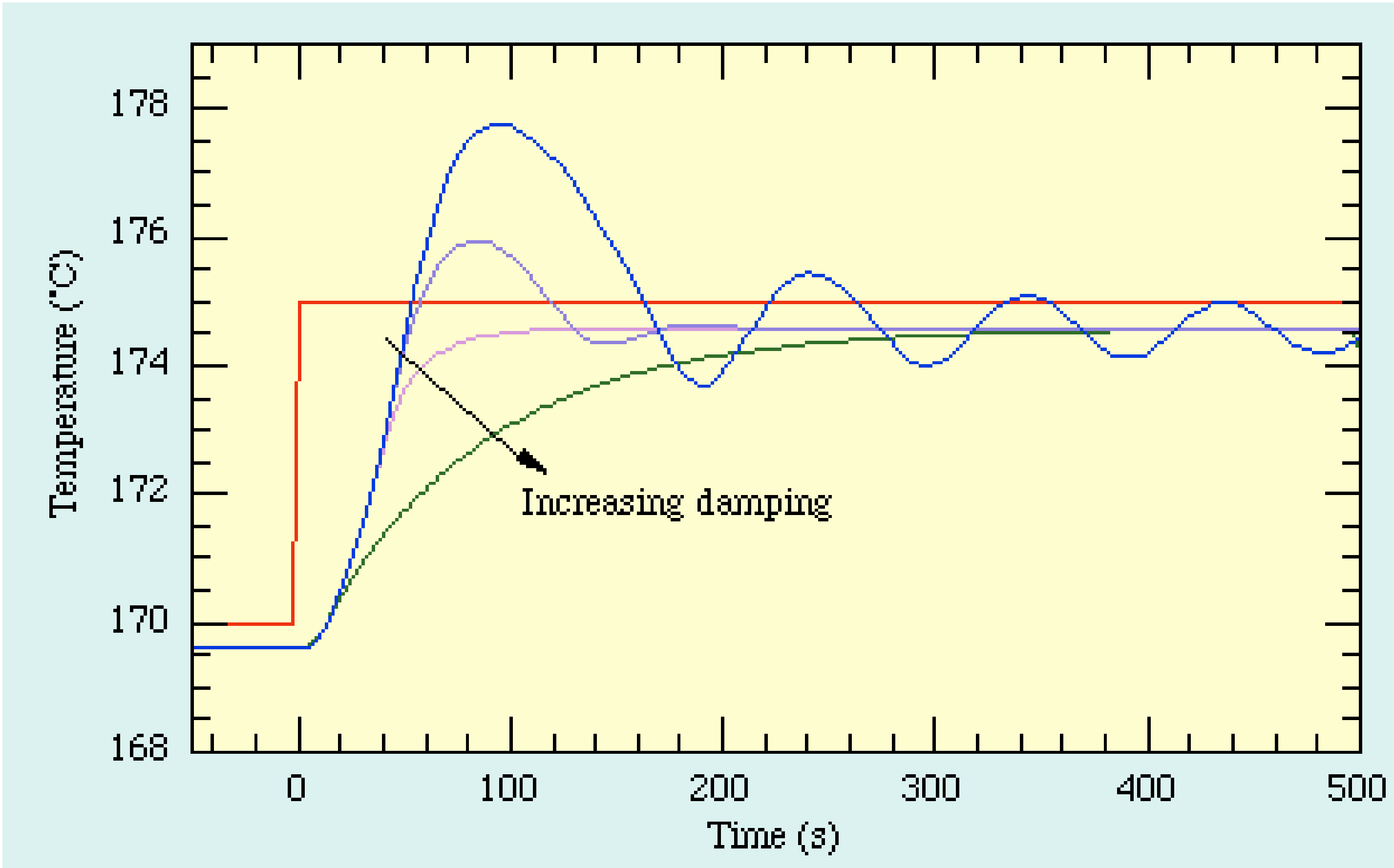
Derivative Control

- Damping friction is a force opposing motion, proportional to velocity.
- Try to prevent overshoot by damping controller response.

$$u = -k_P e - k_D \dot{e}$$

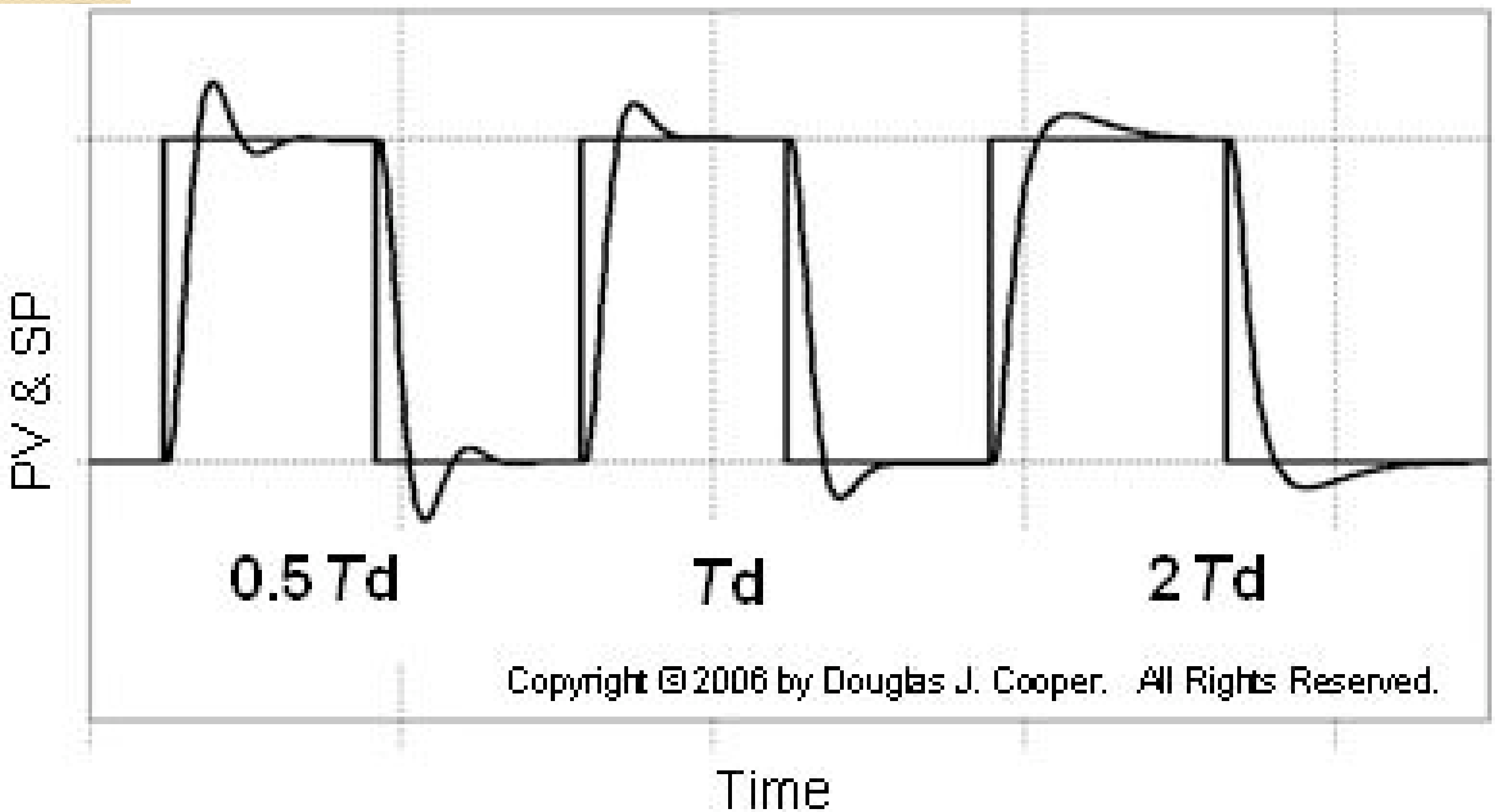
- Estimating a derivative from measurements is fragile, and amplifies noise.

Derivative Control in Action



- Damping fights oscillation and overshoot
- But it's vulnerable to noise

Effect of Derivative Control



- Different amounts of damping (without noise)

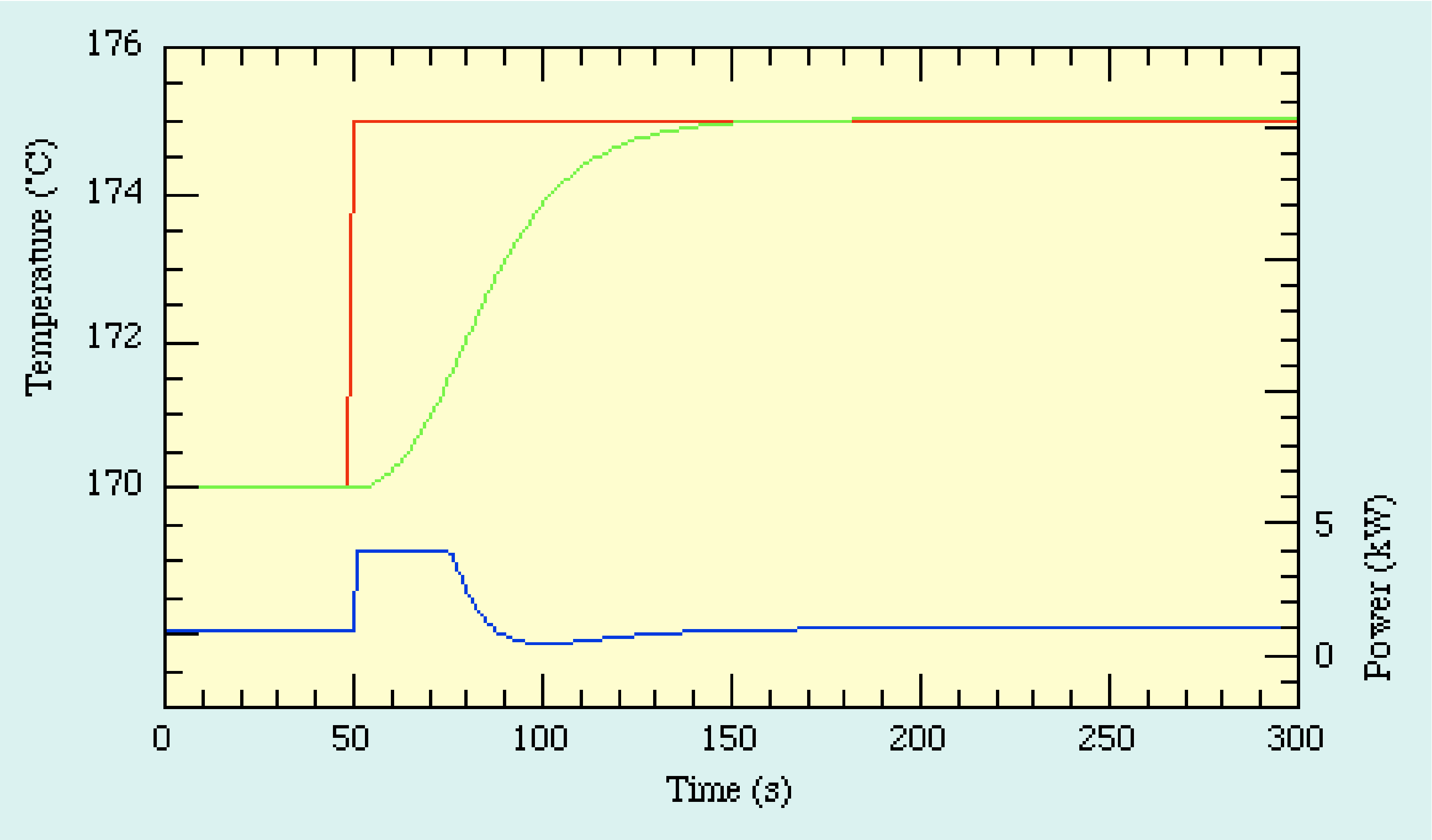
The PID Controller

- A weighted combination of Proportional, Integral, and Derivative terms.

$$u(t) = -k_P e(t) - k_I \int_0^t e dt - k_D \dot{e}(t)$$

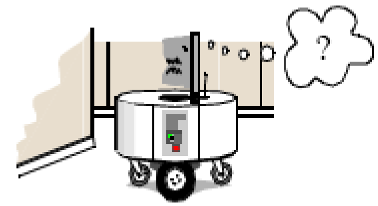
- The PID controller is the workhorse of the control industry. Tuning is non-trivial.

PID Control in Action



- But, good behavior depends on good tuning!

Robot Ingredients



Where am I?
Where am I going?
How do I get there?

- Perception / Sensing
  - Sense external world
- Localization / Estimation
  - Figure out where I am.
- Control
  - Take an action. (How to reach desired state.)

- Planning
  - Given knowledge of external world (from perception/sensing) and myself (localization), what series of control actions must a robot take.

	Rovio	Arm	Heli-copter	Car	NI-Robot
Perception					
Localization					
Control					
Planning					

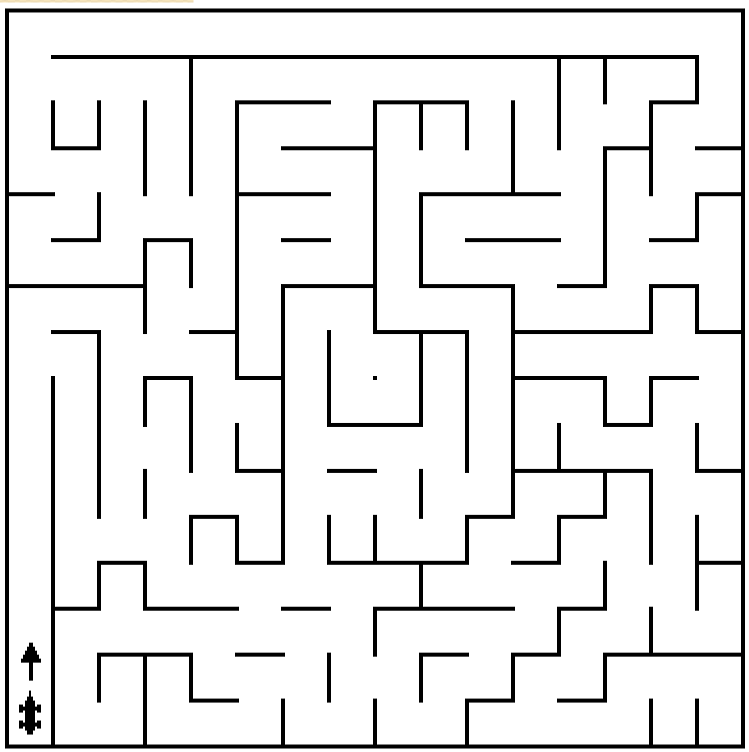
Followers

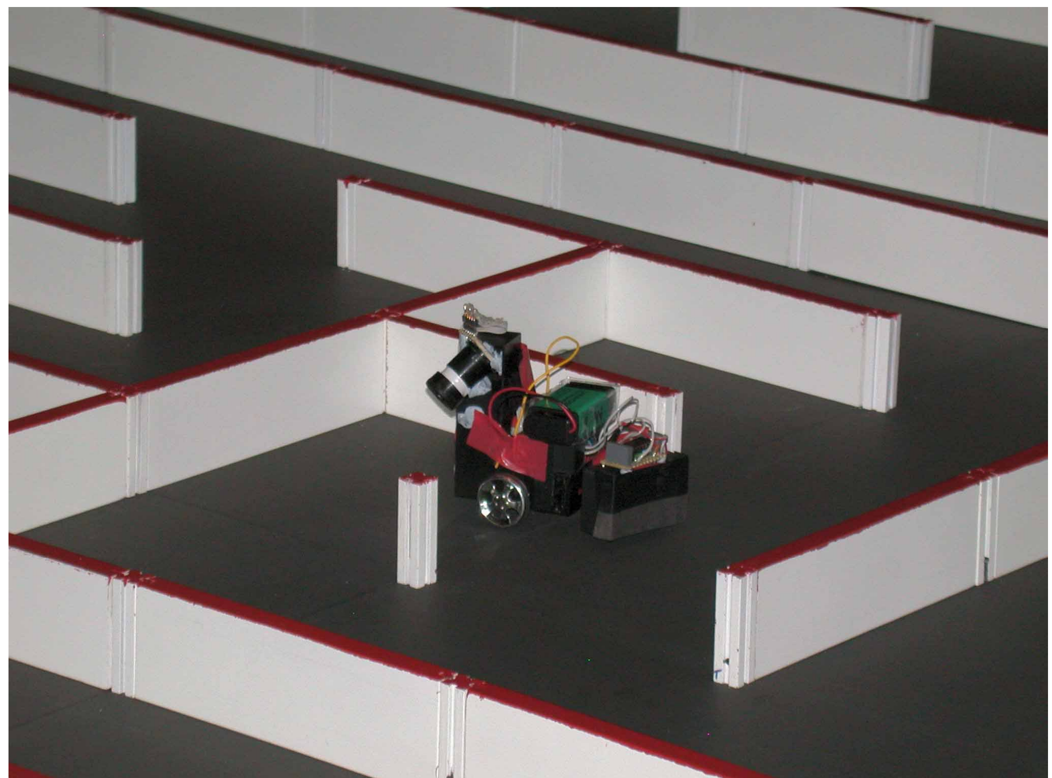
- Very basic plan.

- A follower is a control law where the robot moves forward while keeping some error term small.
  - Open-space follower
  - Wall follower
  - Coastal navigator
  - Color follower

Wall Follower

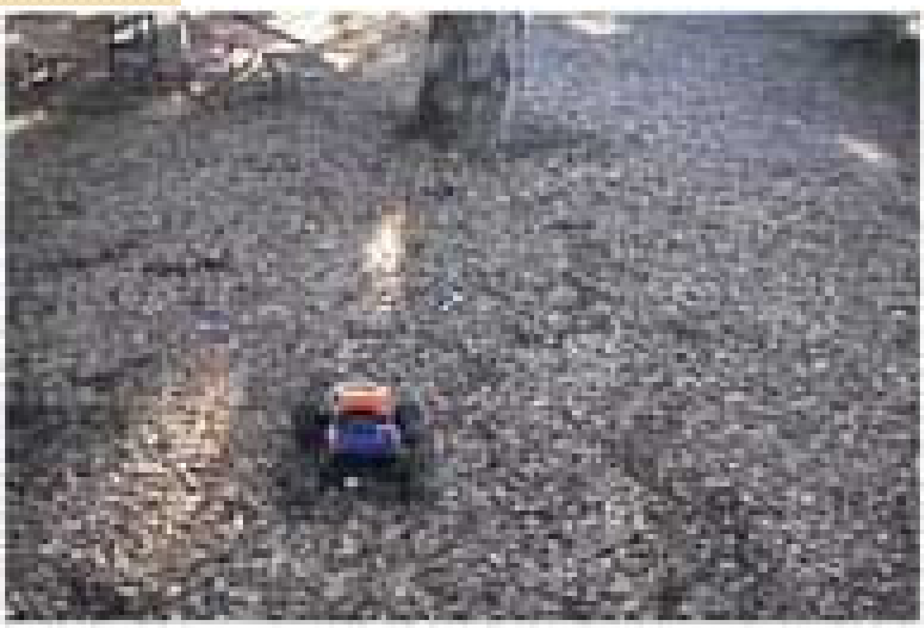
- Detect and follow right or left wall.
- PD control law.
- Tune to avoid large oscillations.





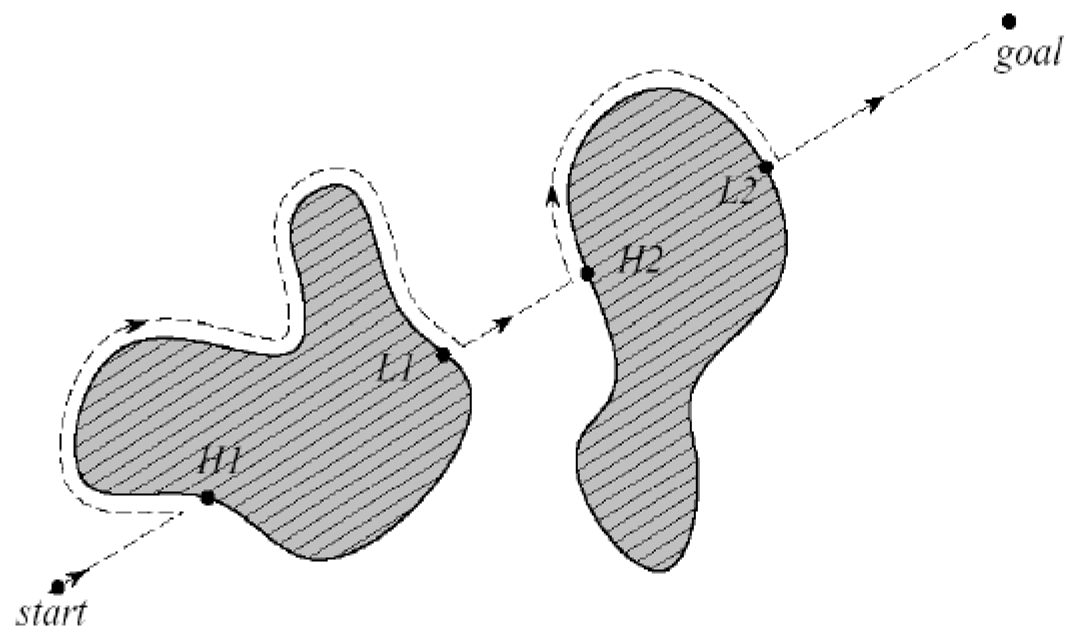
Open-Space Follower

- Move in the direction of large amounts of open space.
- Turn away from obstacles.
- Turn or back out of blind alleys.



Coastal Navigator

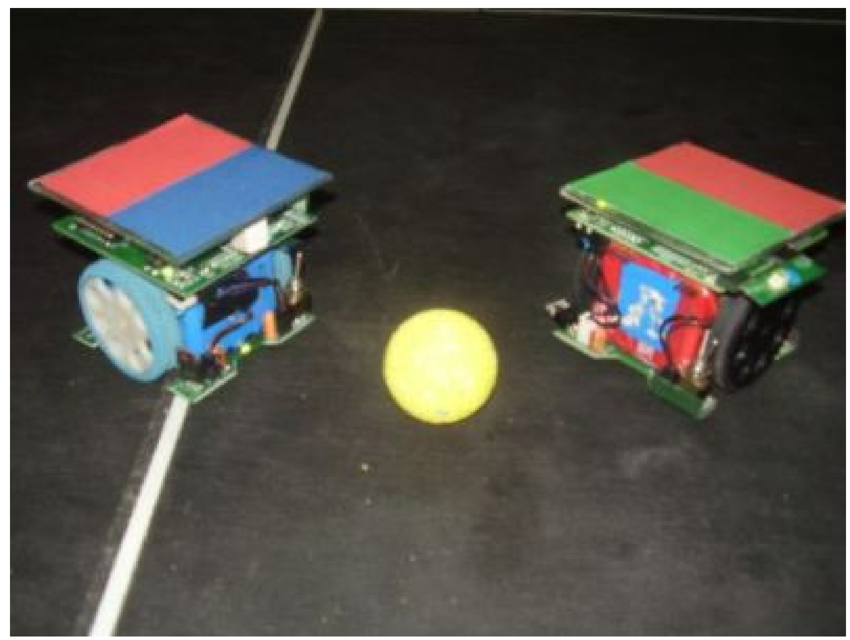
- Join wall-followers to follow a complex "coastline"
- When a wall-follower terminates, make the appropriate turn, detect a new wall, and continue.
- Inside and outside corners, 90 and 180 deg.
- Orbit a box, a simple room, or the desks.



Color Follower

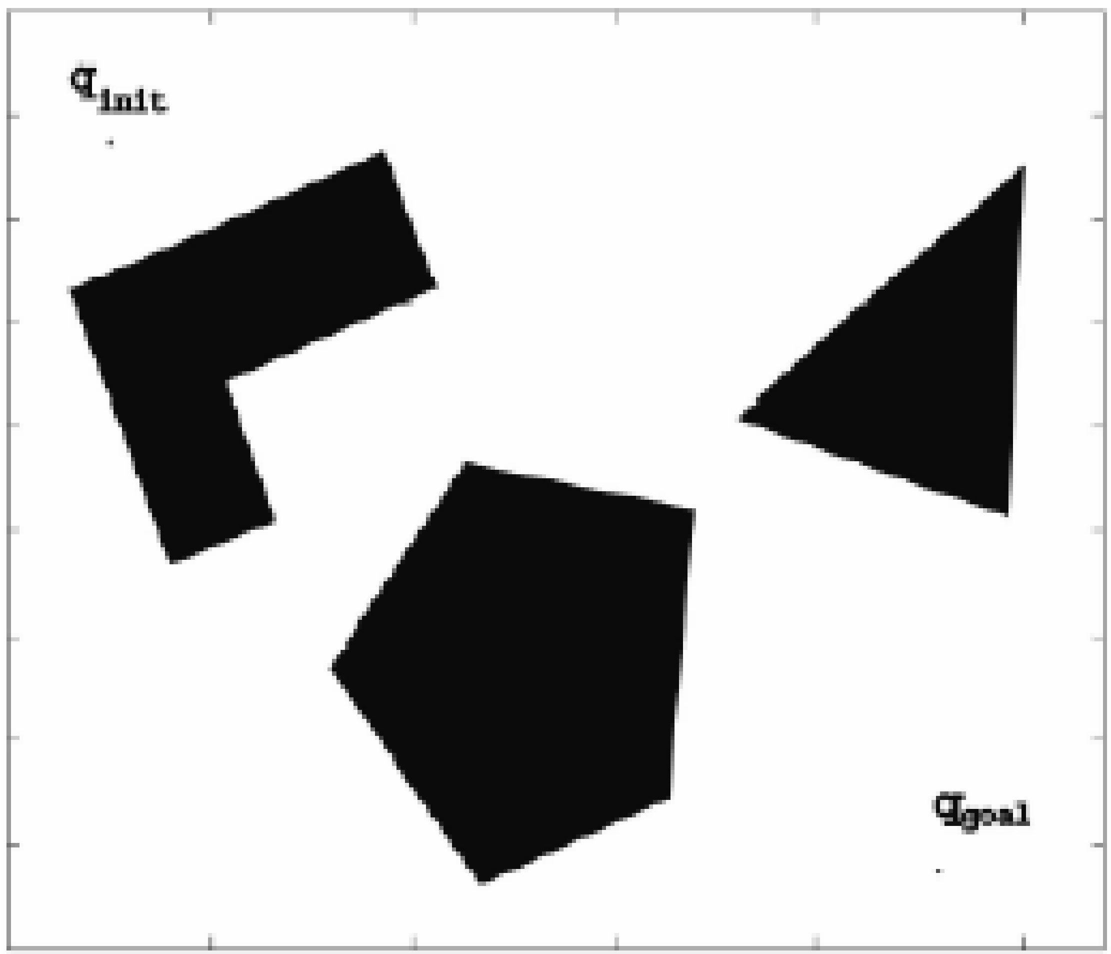
- Move to keep a desired color centered in the camera image.
- Train a color region from a given image.
- Follow an orange ball on a string, or a brightly-colored T-shirt.
- A special case of

"visual servoing."



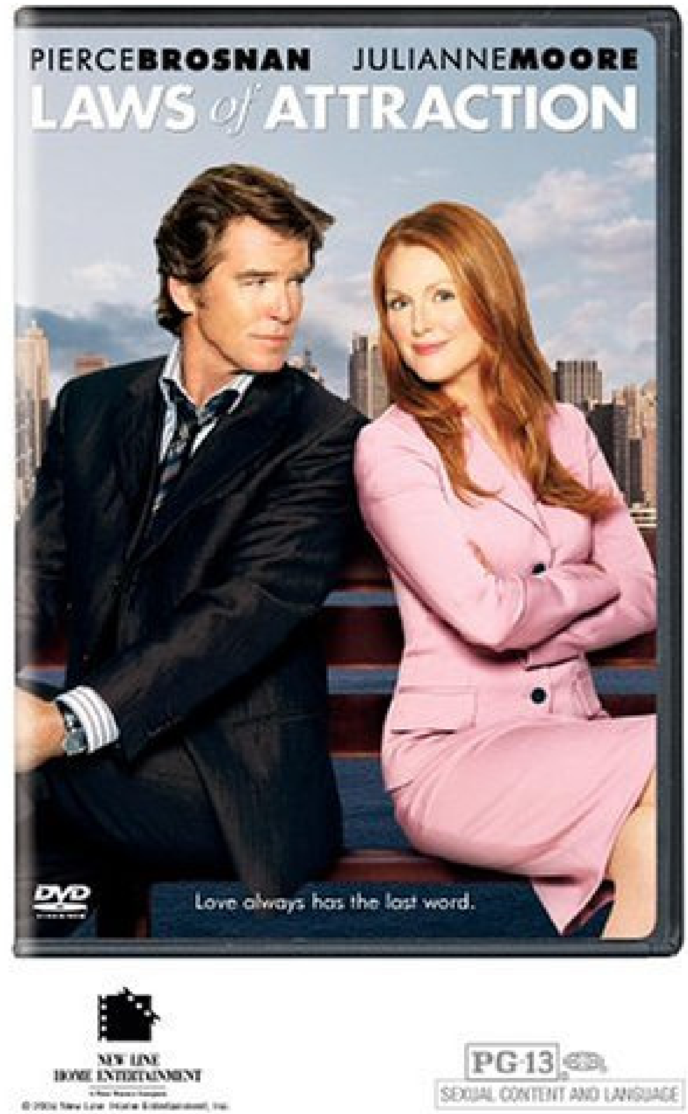
Path Planning

When you know the map, and want to plan a reasonably optimal path.

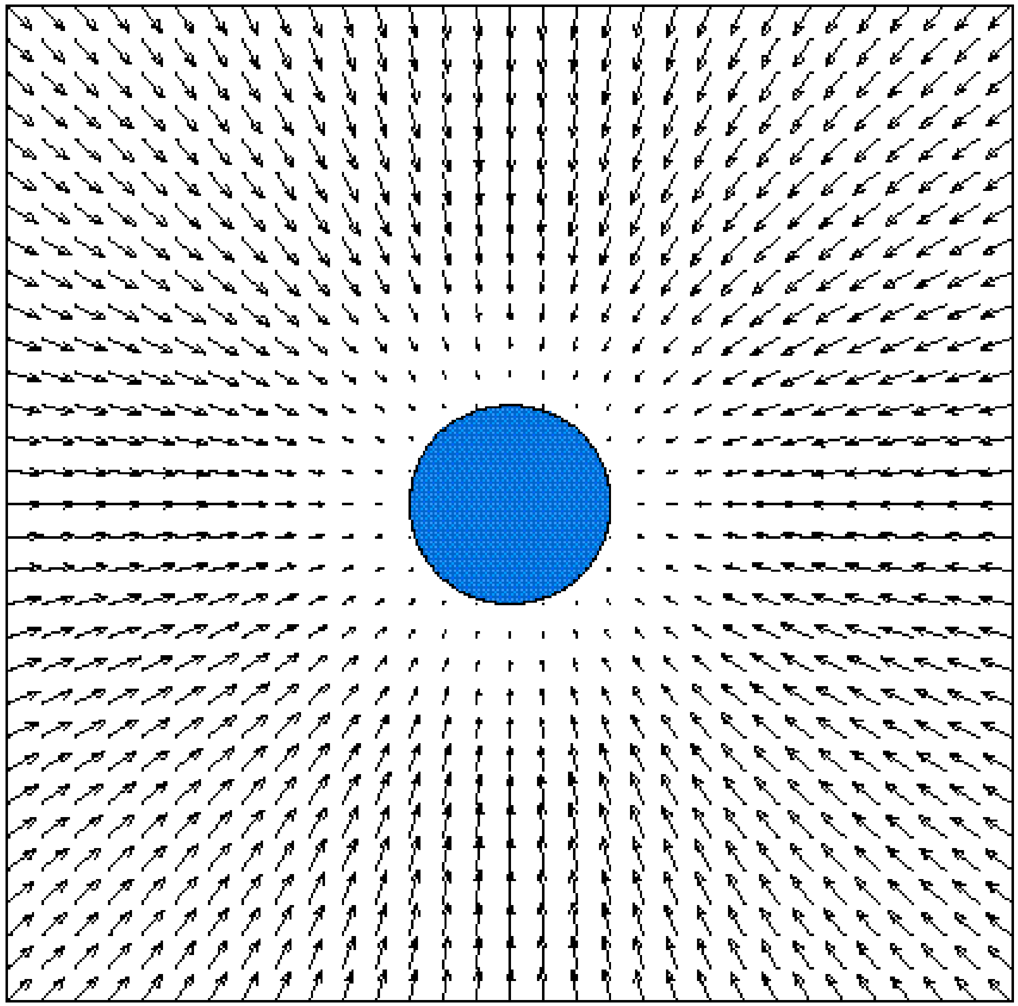


Potential Field

- Its all about laws of "attraction."
- And "repulsion" as well.

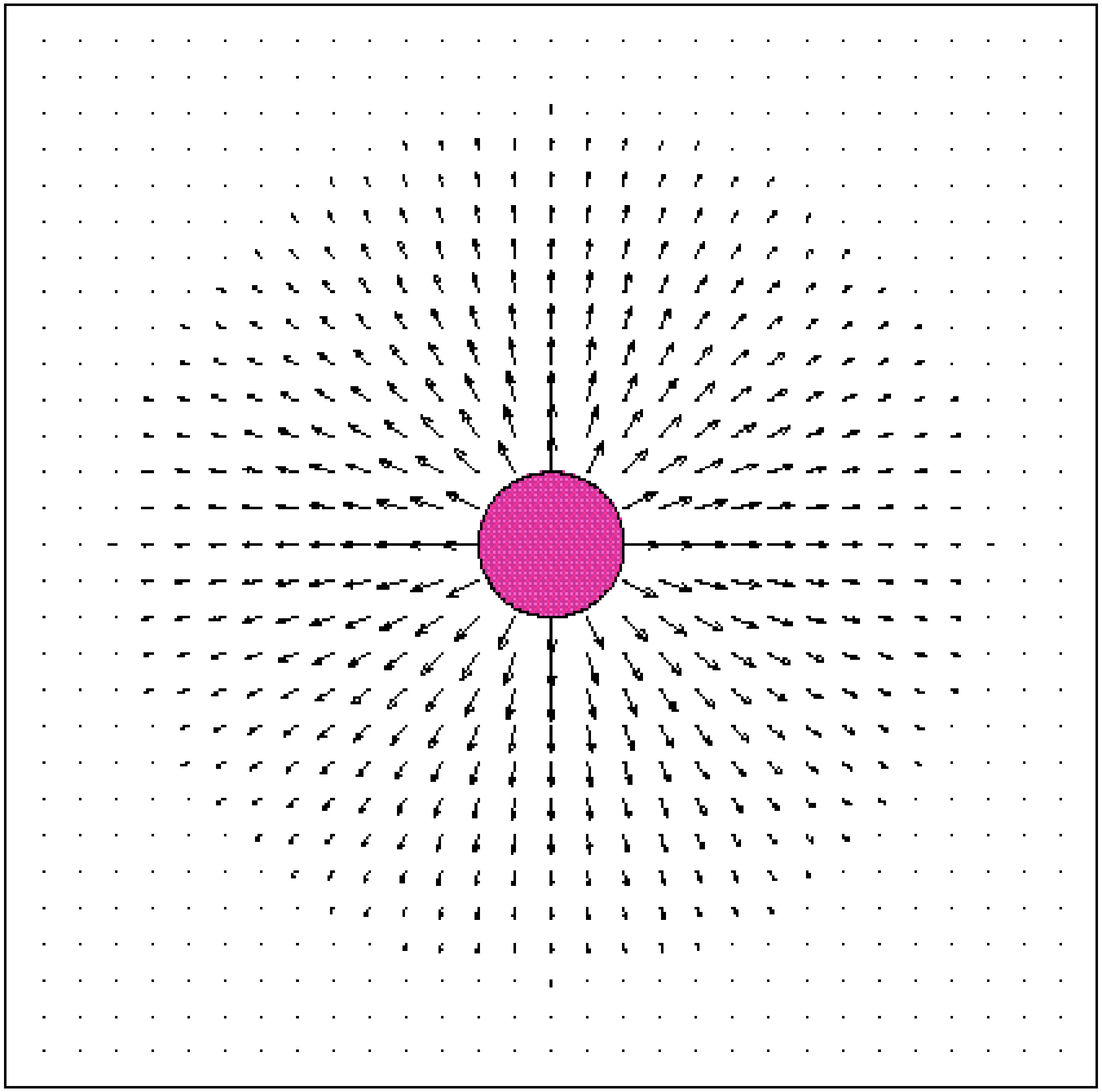


Attractive Potential Field



- Attraction potential: $\frac{1}{2} k_{att} \cdot (q - q_{goal})^2$

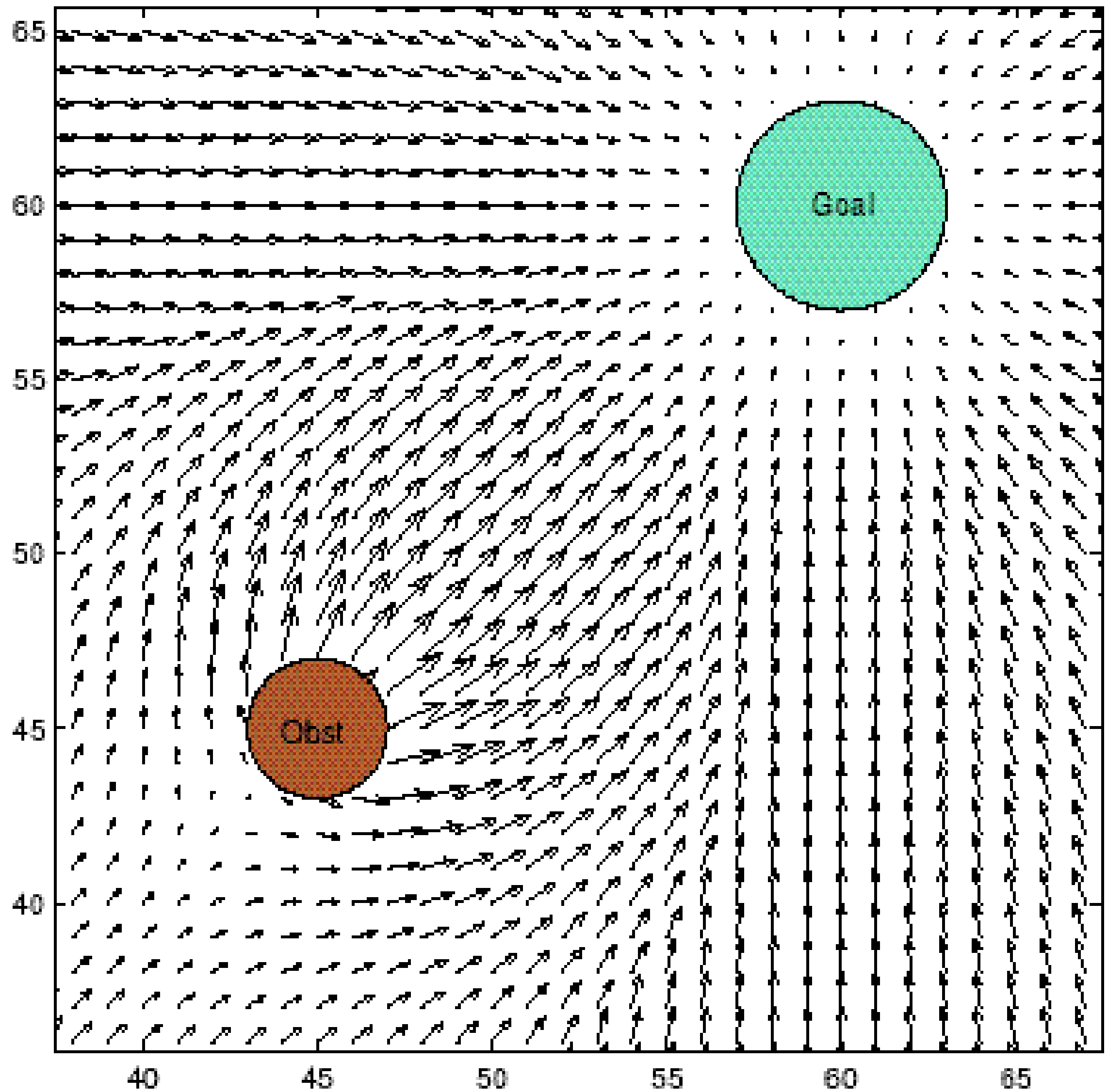
Repulsive Potential Field



- Repulsive Potential

$$= \begin{cases} \frac{1}{2} k_{rep} \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right)^2 & \text{if } \rho(q) \le \rho_0 \\ 0 & \text{if } \rho(q) \ge \rho_0 \end{cases}$$

Vector Sum of Two Fields

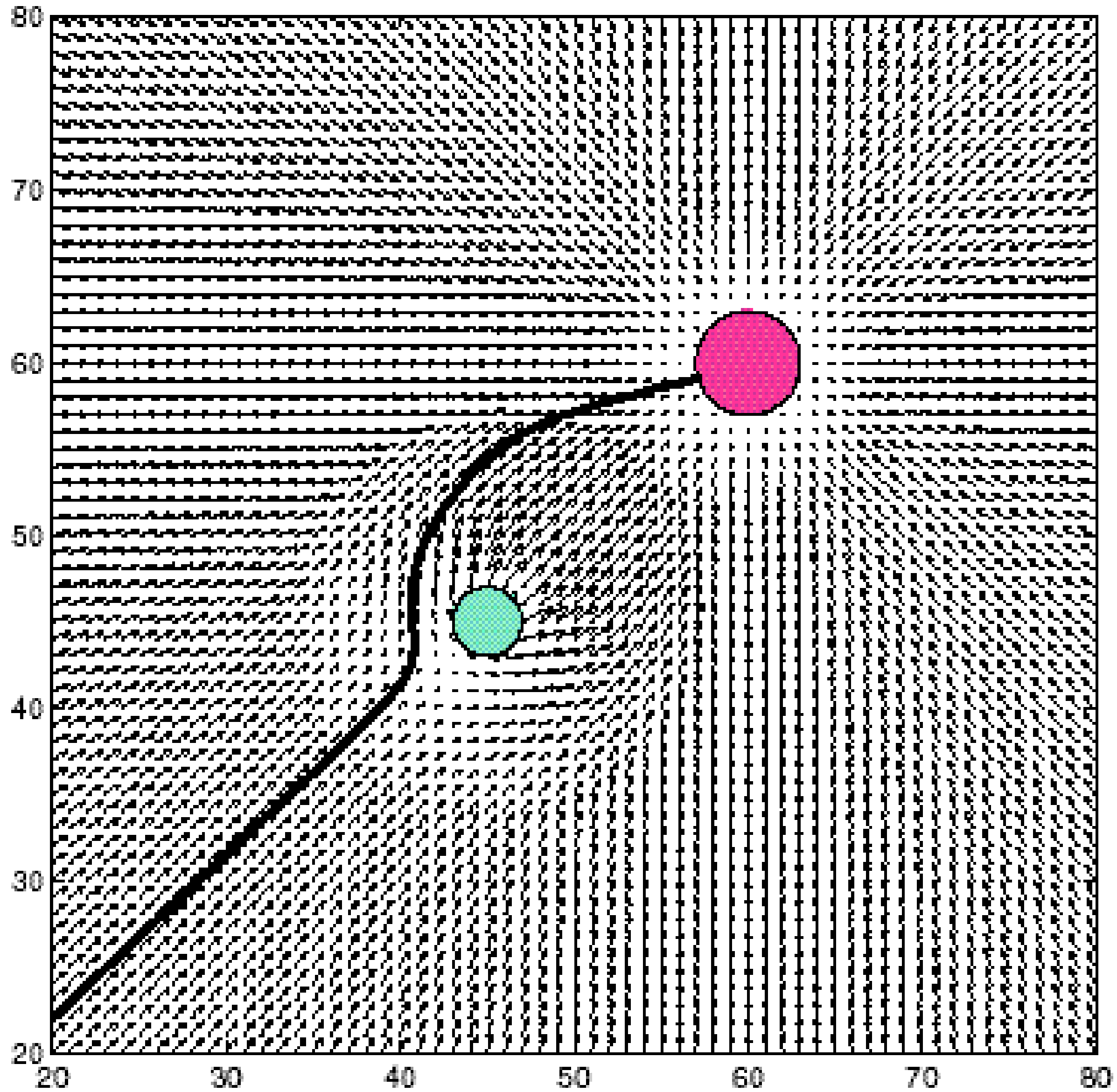


- Generate artificial force field *F(q)*

$$F(q) = -\nabla U(q) = -\nabla U_{att}(q) - \nabla U_{rep}(q) = \begin{bmatrix} \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \end{bmatrix}$$

- Set robot speed (v_x, v_y) proportional to the force $F(q)$ generated by the field
  - the force field drives the robot to the goal
  - robot is assumed to be a point mass

Resulting Robot Trajectory



Potential Fields

- Control laws meant to be added together are often visualized as vector fields:

$$(x, y) \rightarrow (\Delta x, \Delta y)$$

- In some cases, a vector field is the *gradient* of a potential function *P*(*x*,*y*):

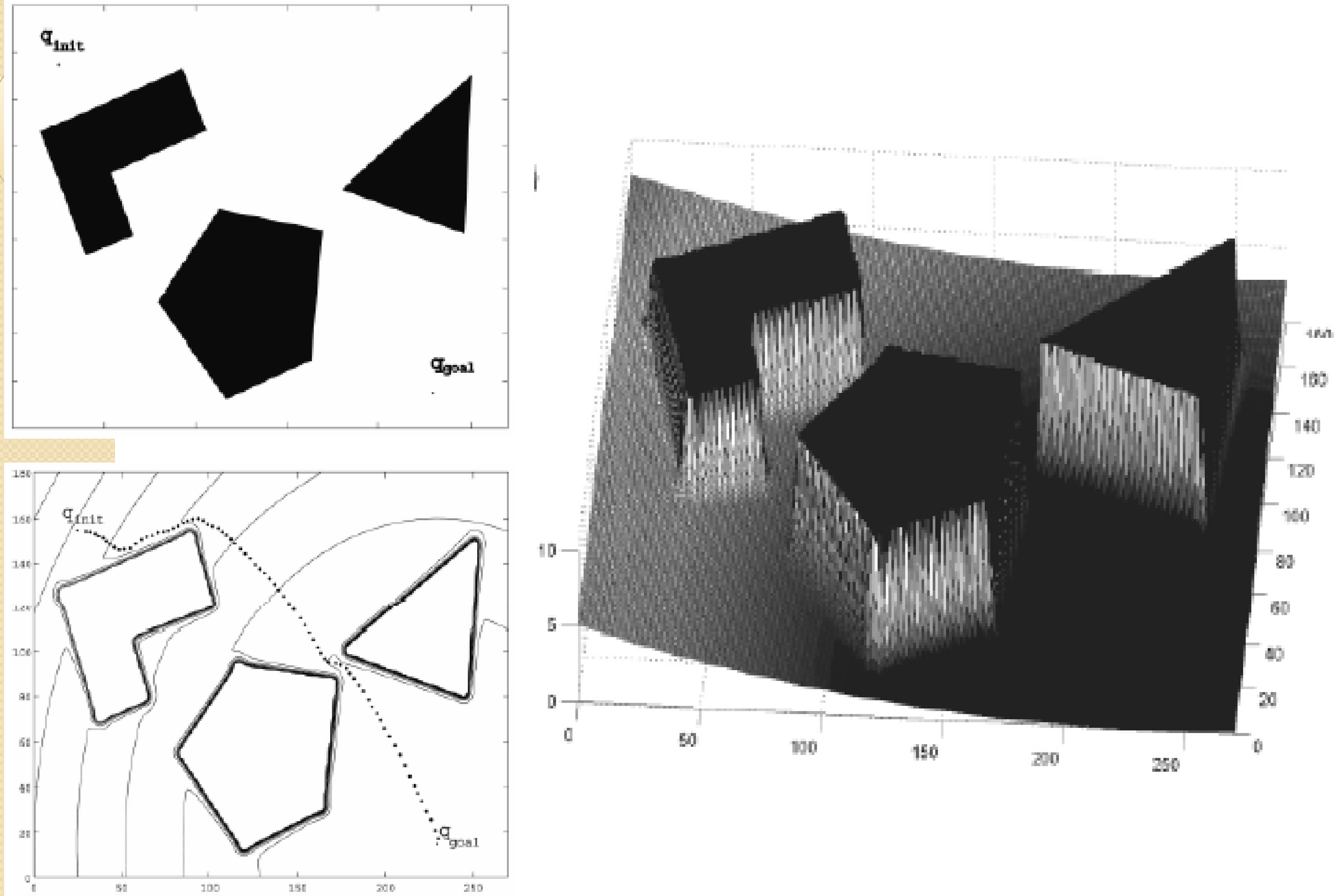
$$(\Delta x, \Delta y) = \nabla P(x, y) = \left(\frac{\partial P}{\partial x}, \frac{\partial P}{\partial y} \right)$$

Potential Fields

- The potential field $P(\mathbf{x})$ is defined over the environment.
- Sensor information $\mathbf{y}$ is used to estimate the potential field gradient $\nabla P(\mathbf{x})$
  - No need to compute the entire field.
  - Compute individual components separately.
- The motor vector $\mathbf{u}$ is determined to follow that gradient.

Attraction and Avoidance

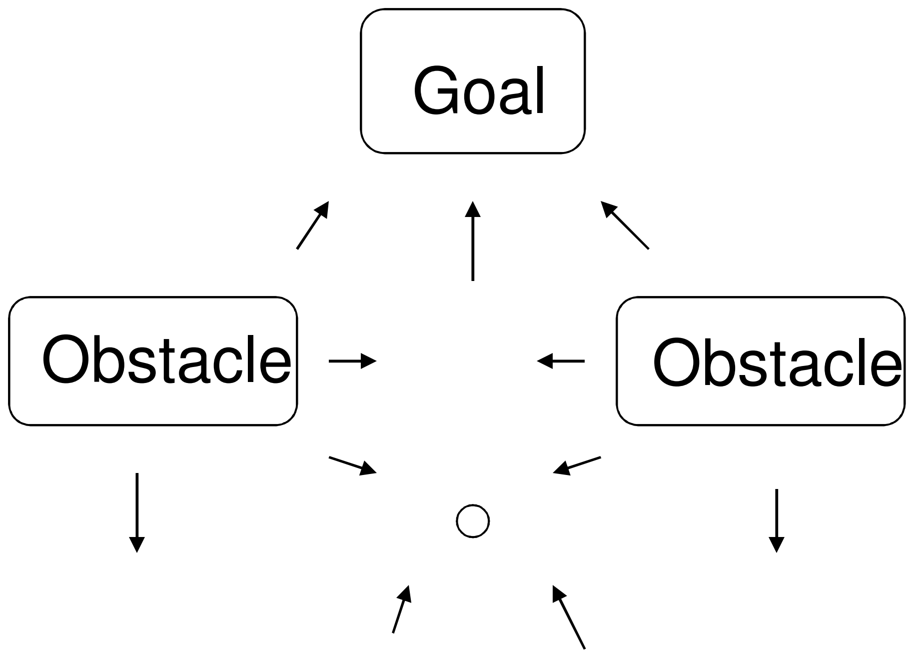
- **Goal**: Surround with an attractive field.
- **Obstacles**: Surround with repulsive fields.
- *Ideal result*: move toward goal while avoiding collisions with obstacles.
  - Think of rolling down a curved surface.
- *Dynamic obstacles*: rapid update to the potential field avoids moving obstacles.


q_init
q_goal
0
50
100
150
200
250
5
10
20
40
60
80
100
120
140
160

Potential Problems with Potential Fields

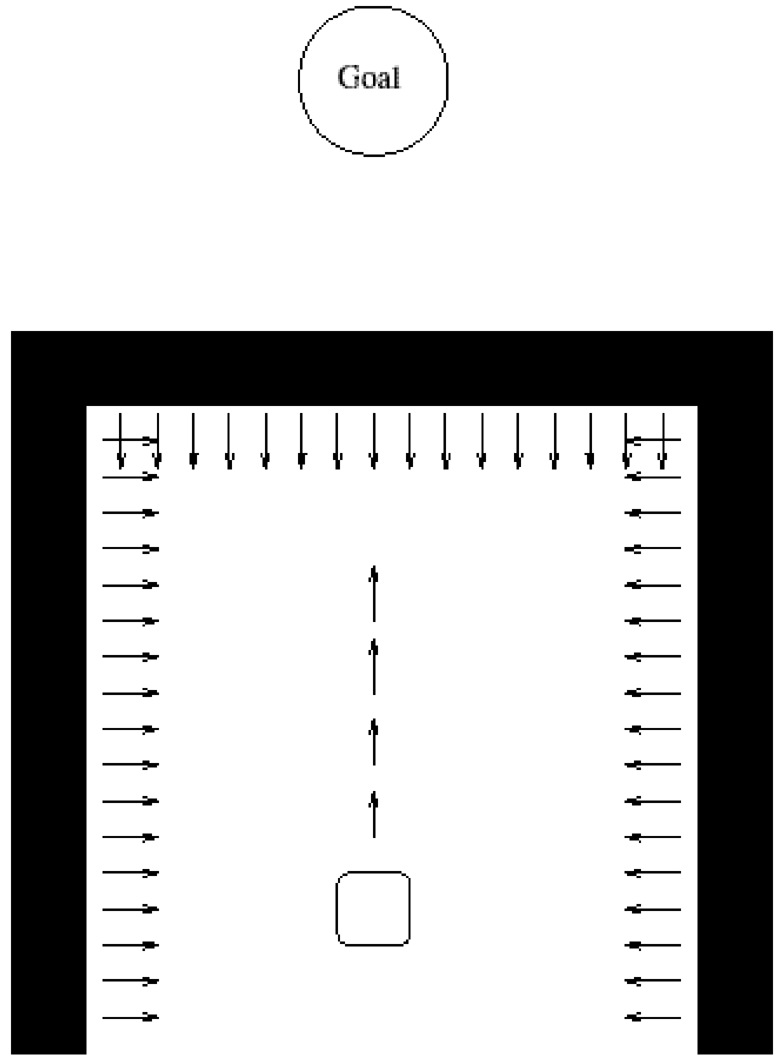
- *Local minima*
  - Attractive and repulsive forces can balance, so robot makes no progress.
  - Closely spaced obstacles, or dead end.
- *Unstable oscillation*
  - The dynamics of the robot/environment system can become unstable.
  - High speeds, narrow corridors, sudden changes.

Local Minimum Problem



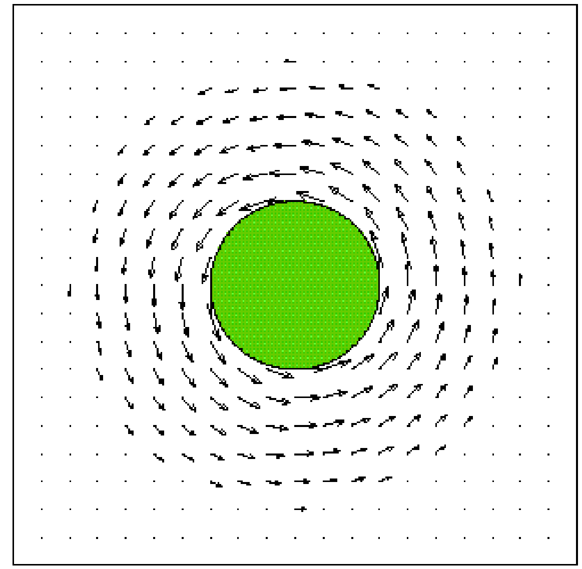
Box Canyon Problem

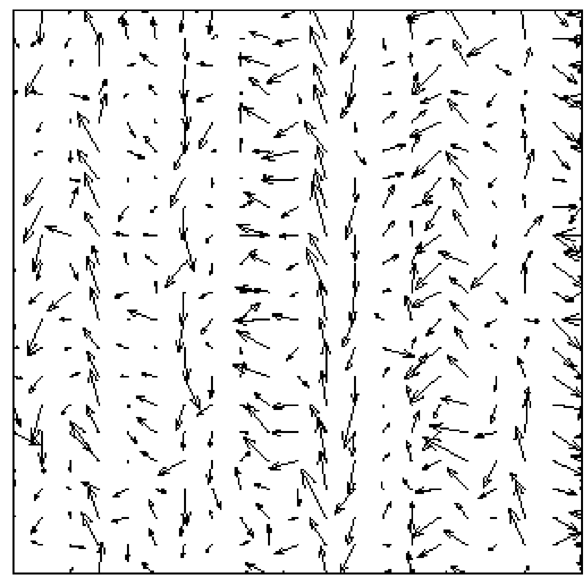
- Local minimum problem, or
- *AvoidPast* potential field.



Rotational and Random Fields

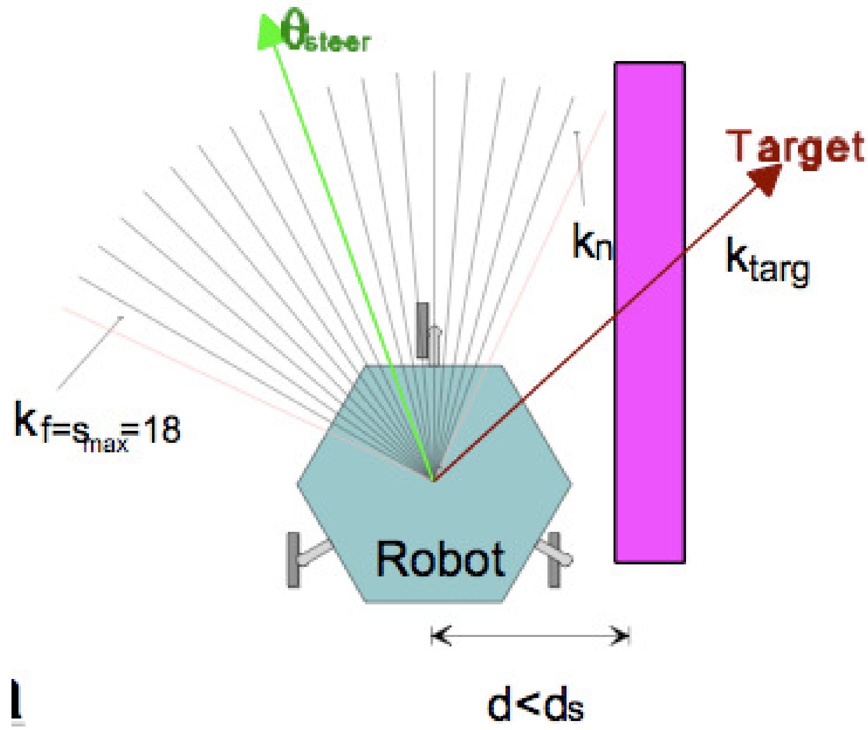
- Not gradients of potential functions
- Adding a *rotational field* around obstacles
  - Breaks symmetry
  - Avoids some local minima
  - Guides robot around groups of obstacles
- A *random field* gets the robot unstuck.
  - Avoids some local minima.

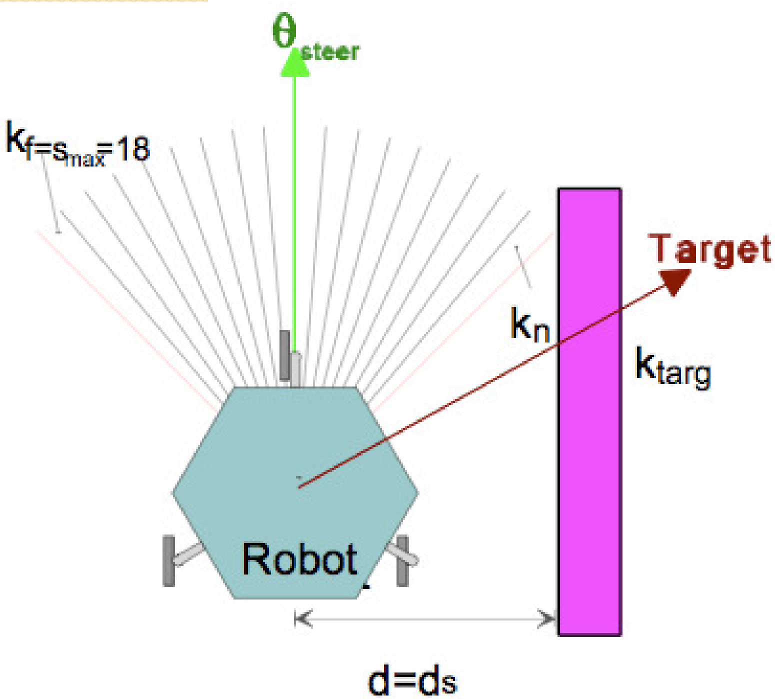


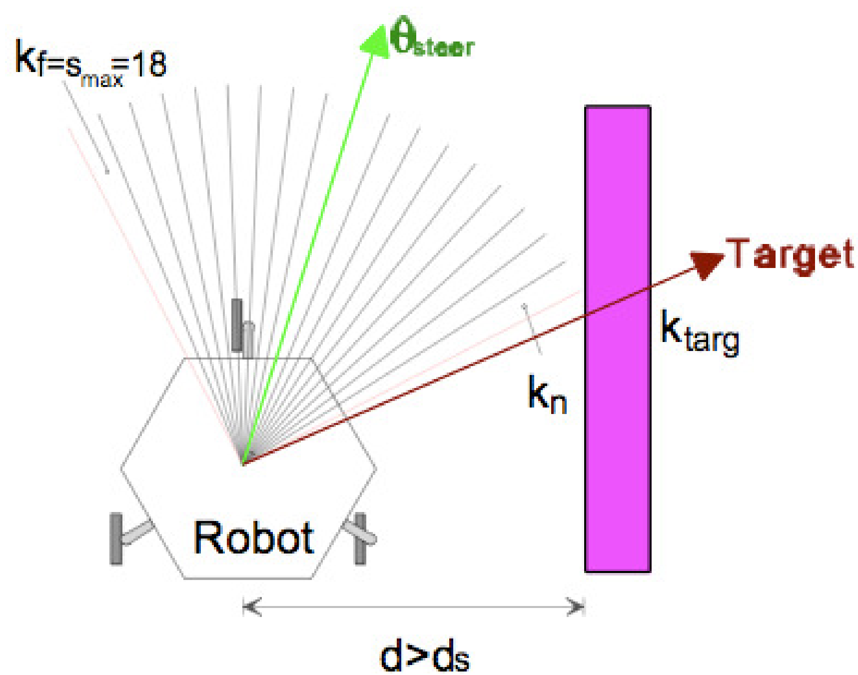


Leads to natural wall-following

- Threshold determines offset from wall.

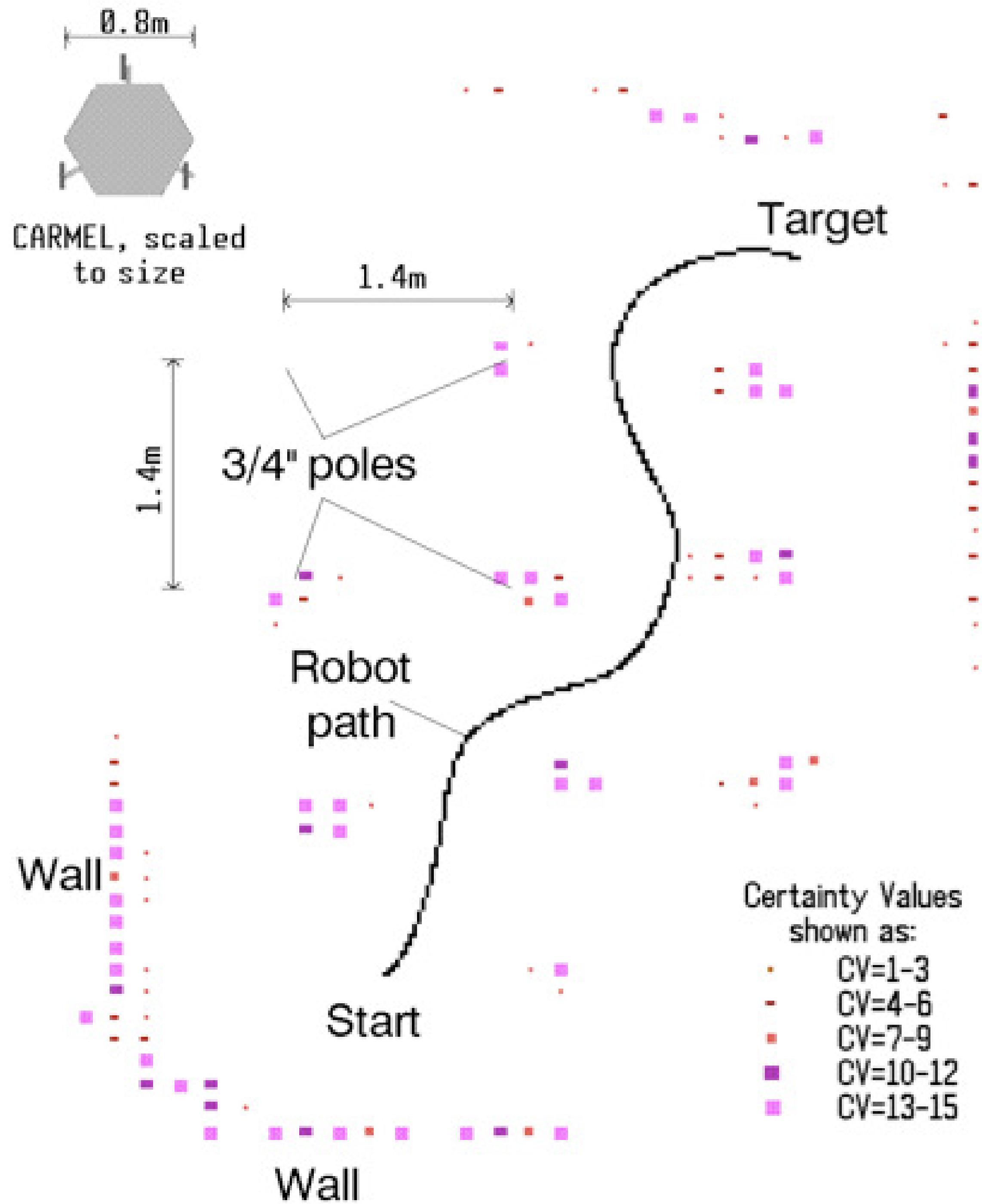






Smooth, Natural Wandering Behavior

- Potentially quite fast!
- 1 m/s or more!

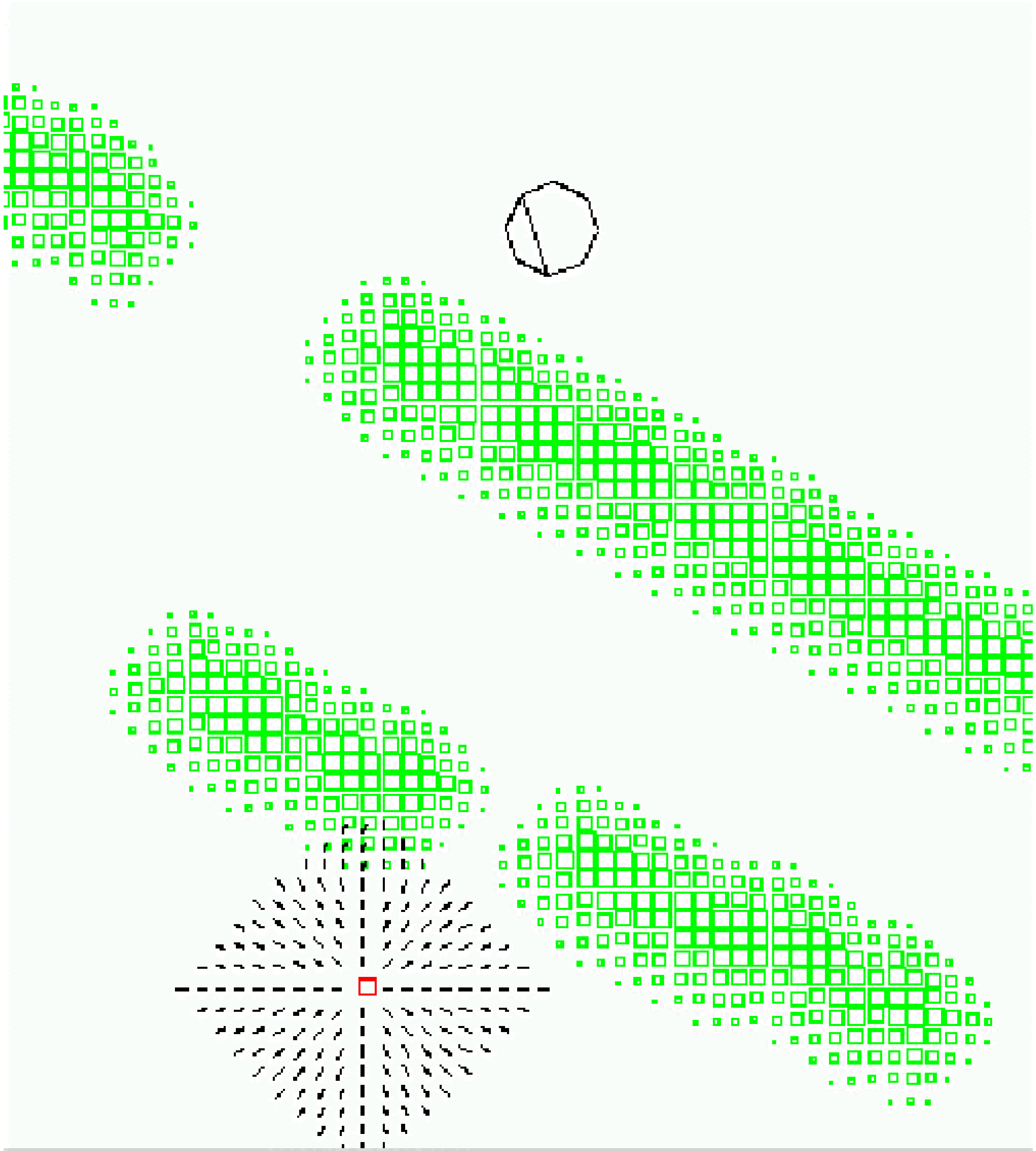


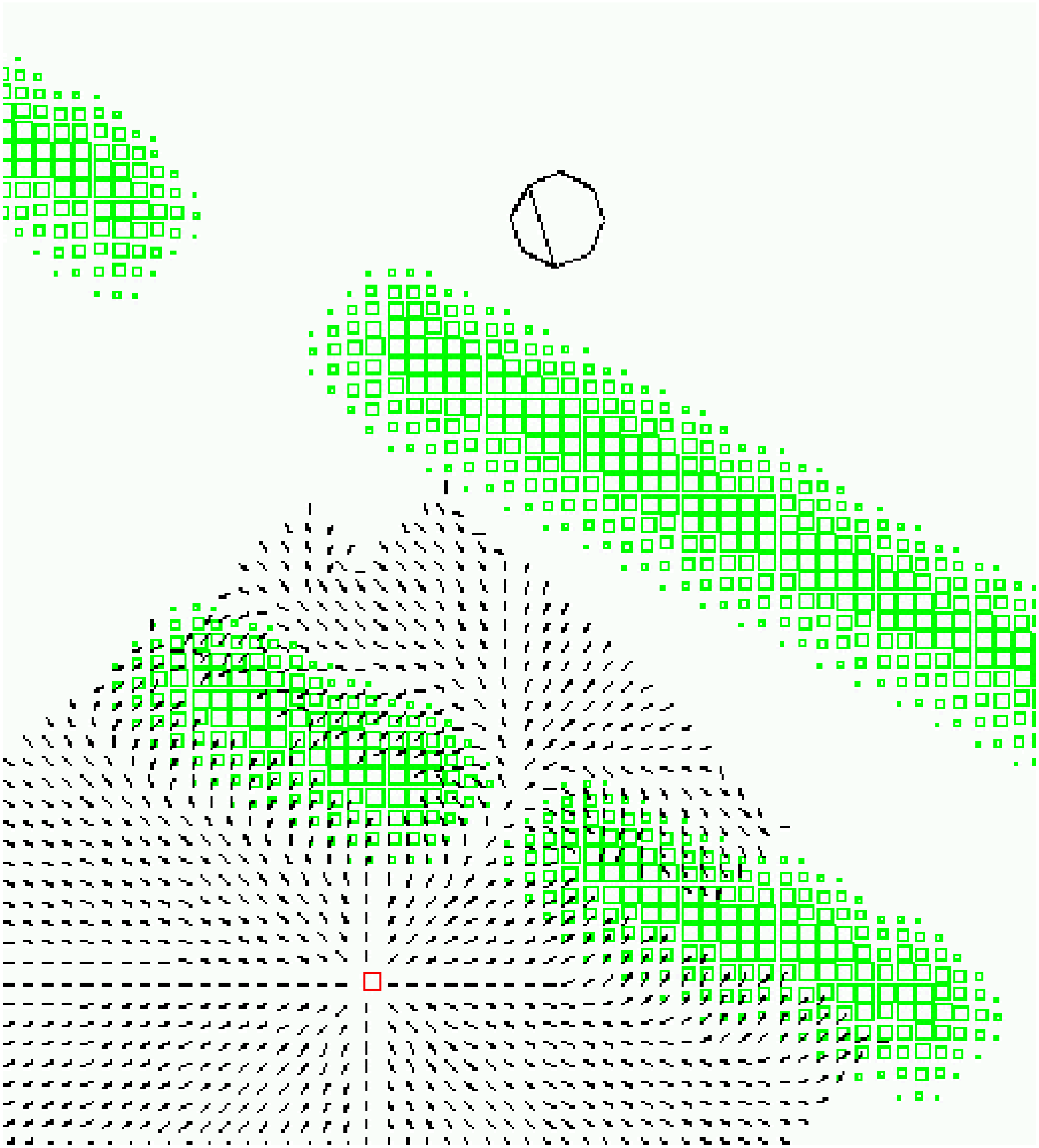
Incorporating path length

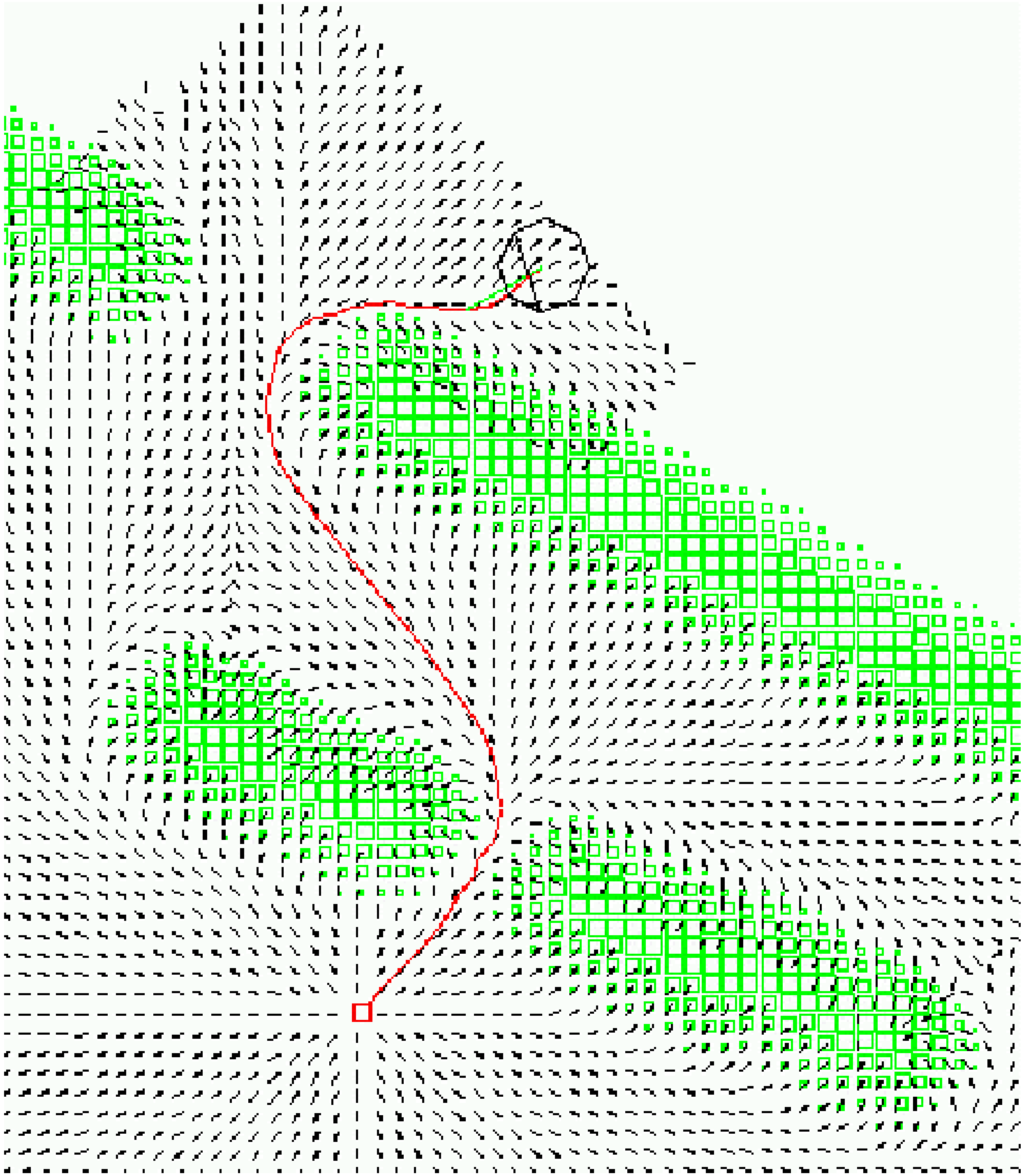
- A *path* is a sequence of points:
  - $P = \{p_1, p_2, p_3, \ldots\}$
- The *cost* of a path is

$$F(P) = \sum_i I(p_i) + \sum_i A(p_i, p_{i+1})$$

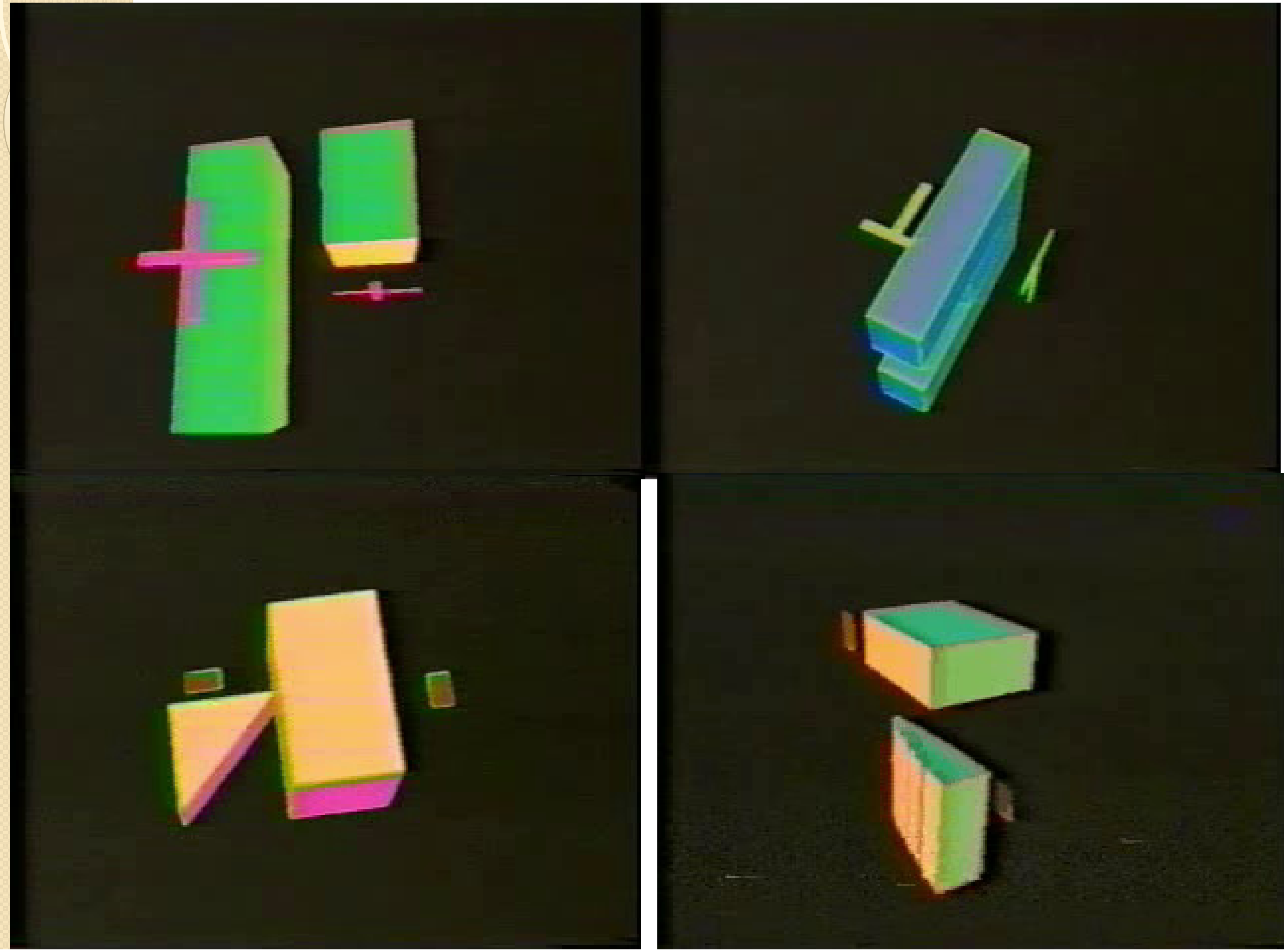
- Intrinsic cost $I(p_i)$ handles obstacles, etc.
- Adjacency cost $A(p_i, p_{i+1})$ handles path length.



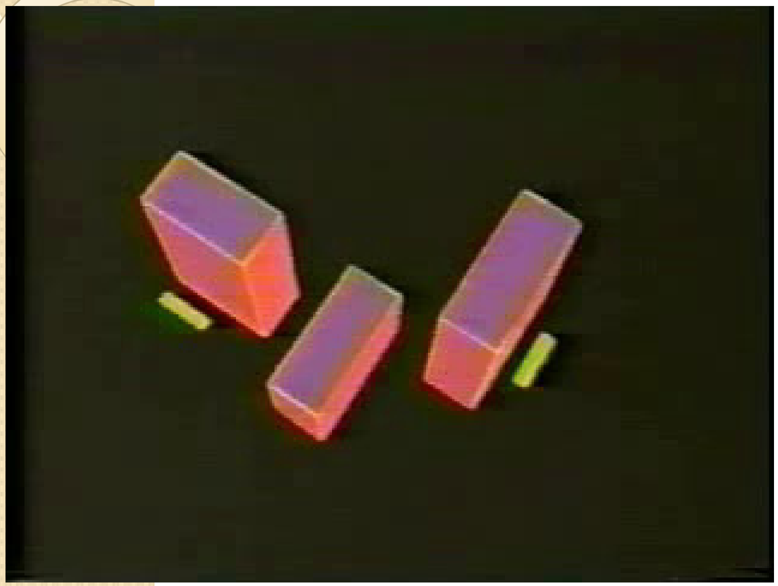


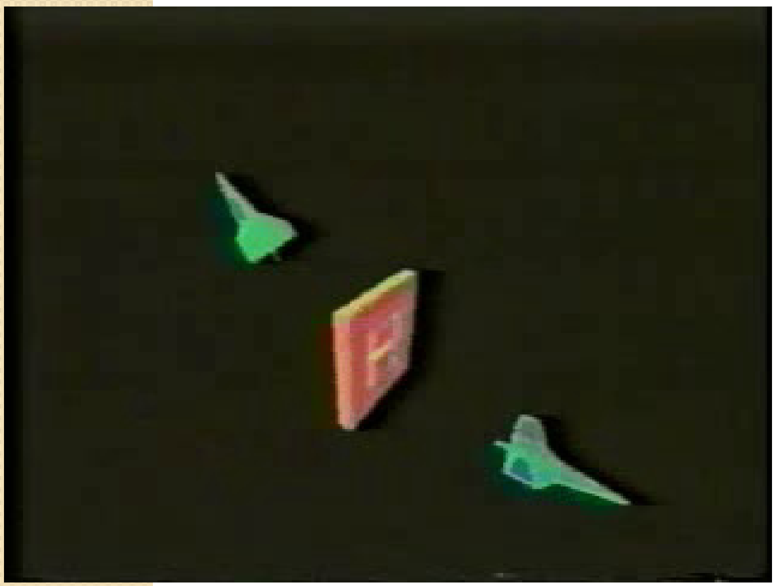


Potential Field Videos

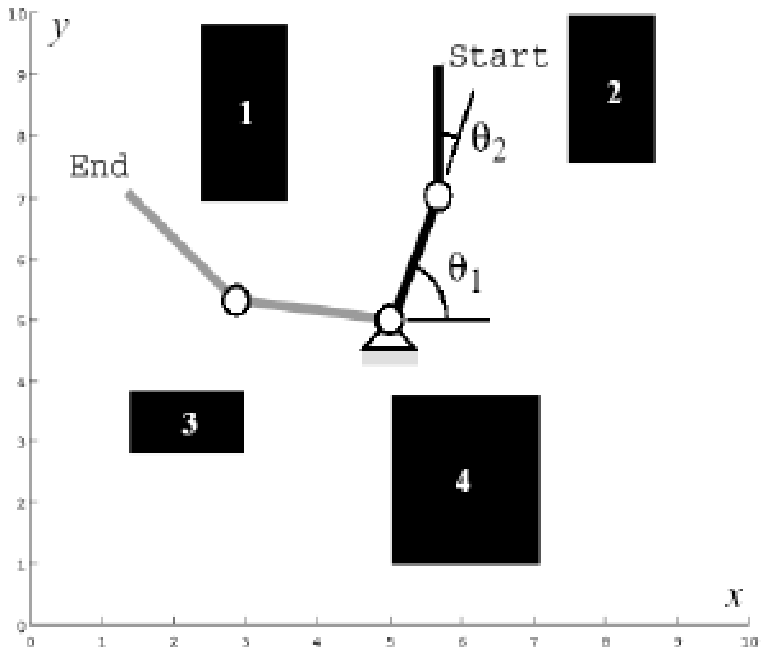


Potential Field Videos

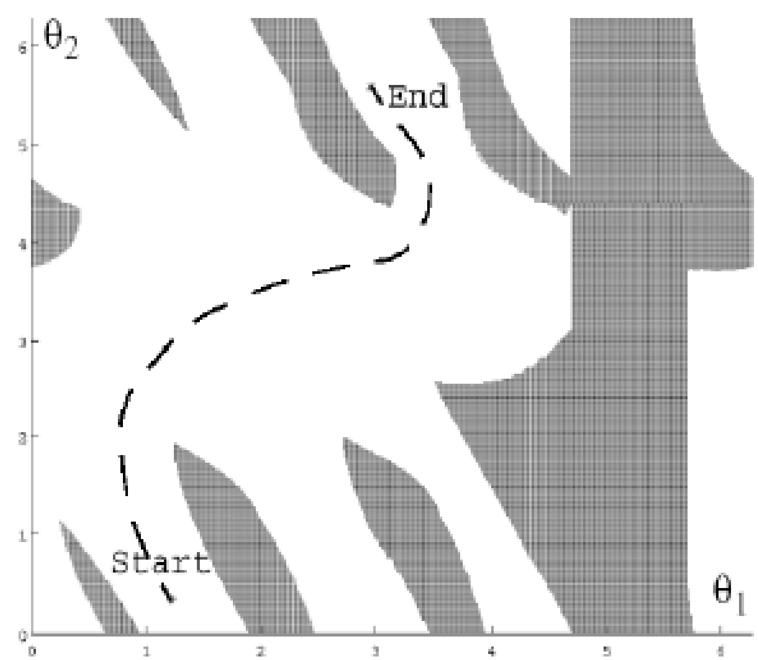




Hard problem for Potential Field



Work Space



Configuration Space:
the dimension of this space is equal to the Degrees of Freedom (DoF) of the robot

Roadmap Path Planning