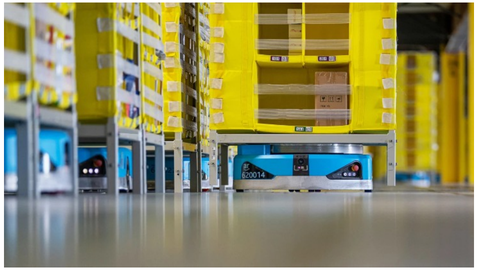
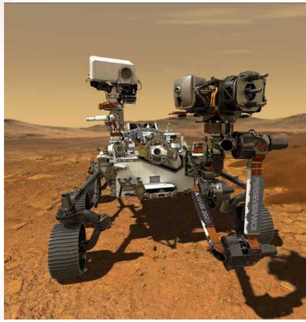
Review

Autonomous robots

- Robot = Programmable machine
- Autonomous = Generate their own behavior based on on-board sensor information
- In practice often a hybrid, low level tasks are done autonomously





Dynamics

Given a state and sensor input tells us how system changes

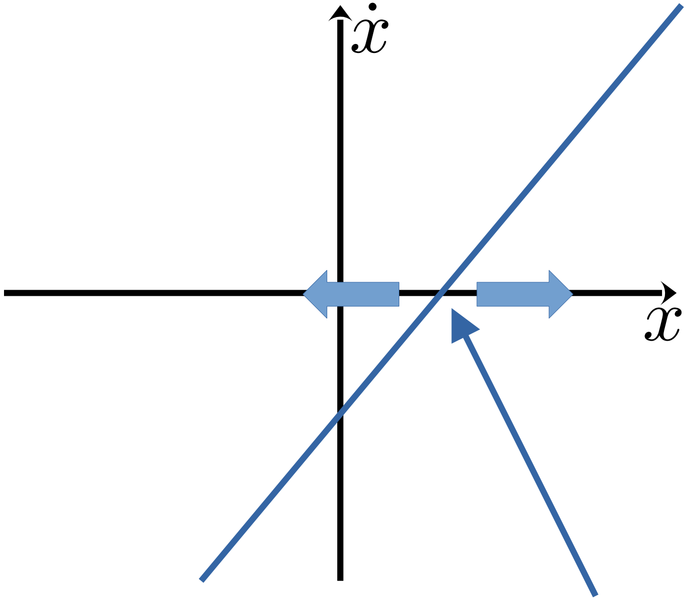
Where does the robot look and what does it sees there

How fast does it turn left or right?

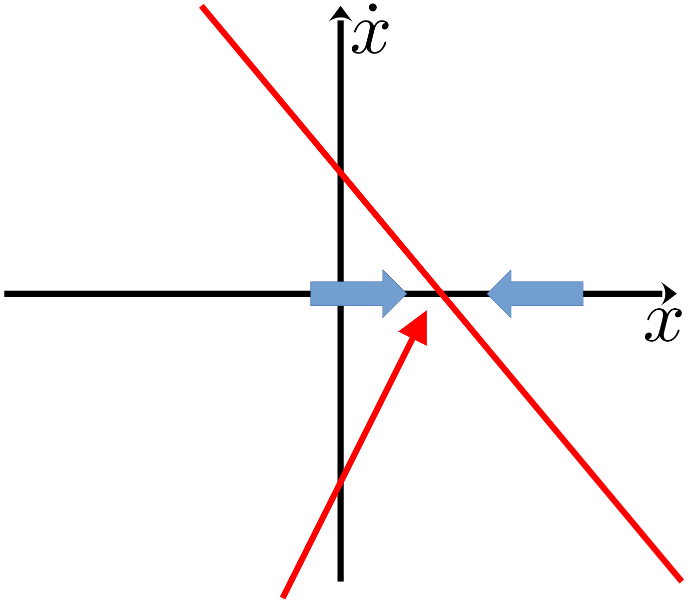
$$f\left(\phi, \mathrm{senordata}\right) \quad = \quad \dot{\phi}$$

<u>Important:</u> Where is $\dot{\phi} = 0$? Where does the robot stop changing?

Attraction and repulsion



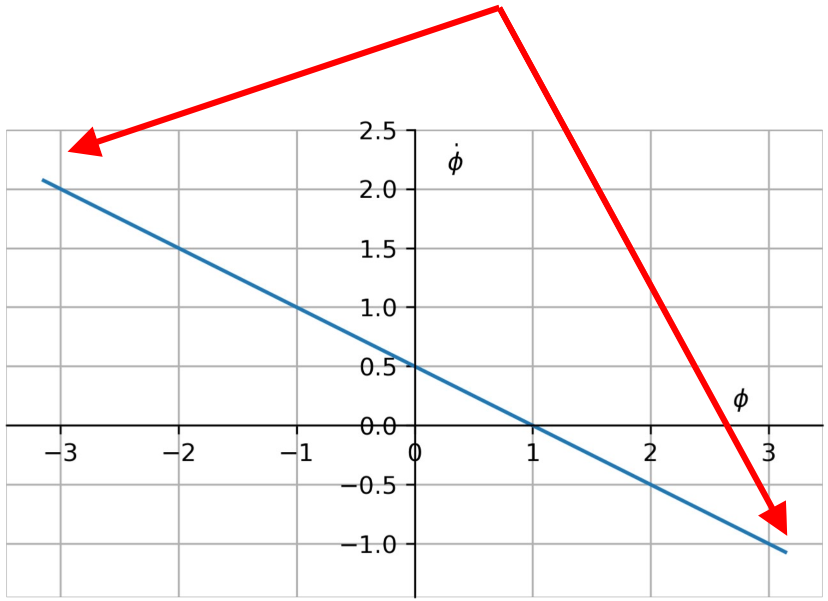
Move away from here

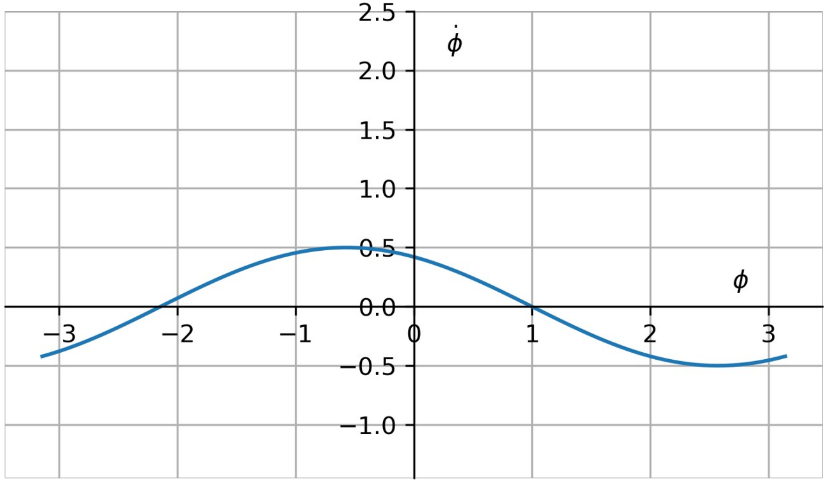


Come here and
stay there

The target attractor

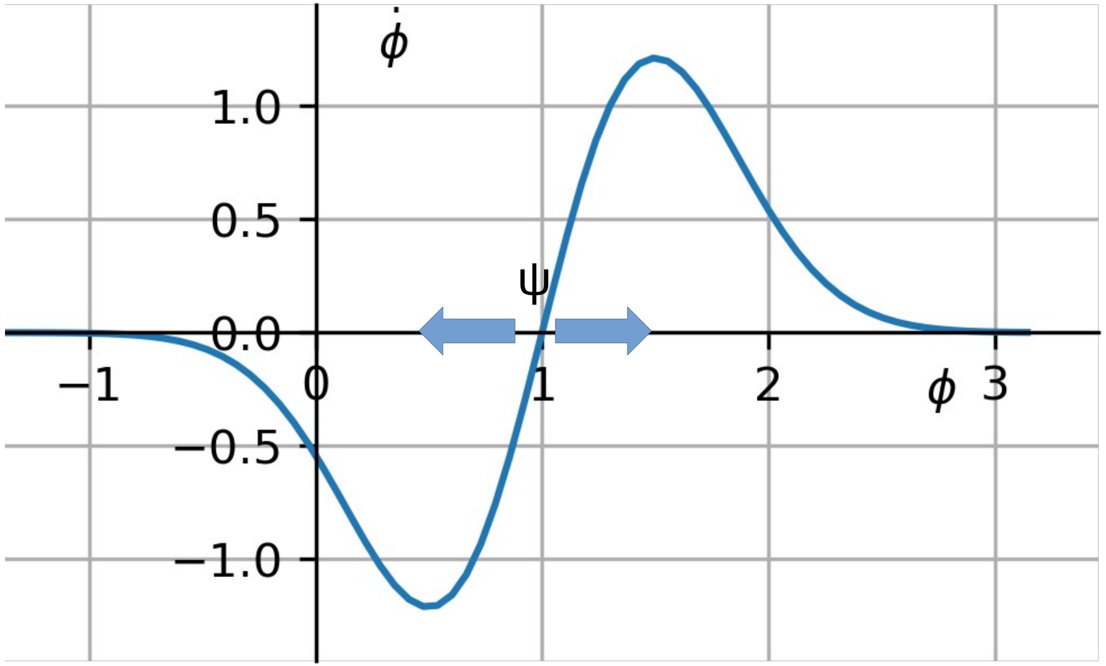
This would cause a very abrupt change
in behavior at the discontinuity





Forcelets for repellors

$$f_i = \lambda_i(\phi - \psi_i)e^{-\frac{(\phi-\psi_i)^2}{2\sigma^2}}$$

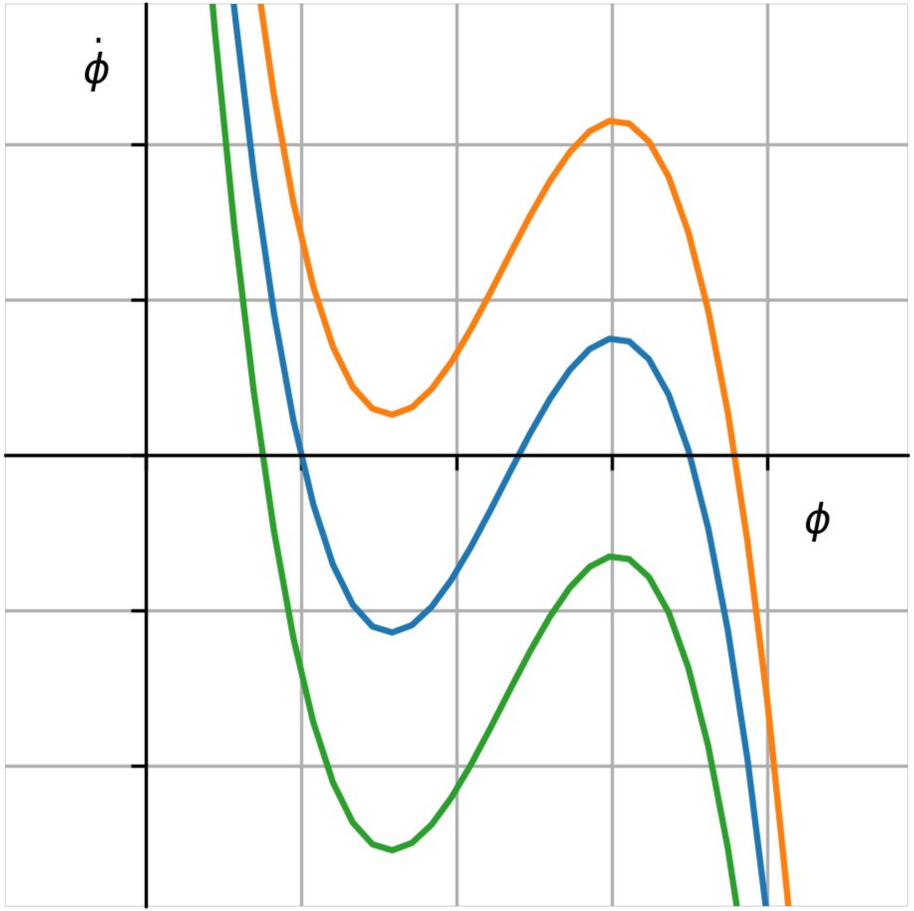


Effectively: Turn away from angle ψ with strength λ

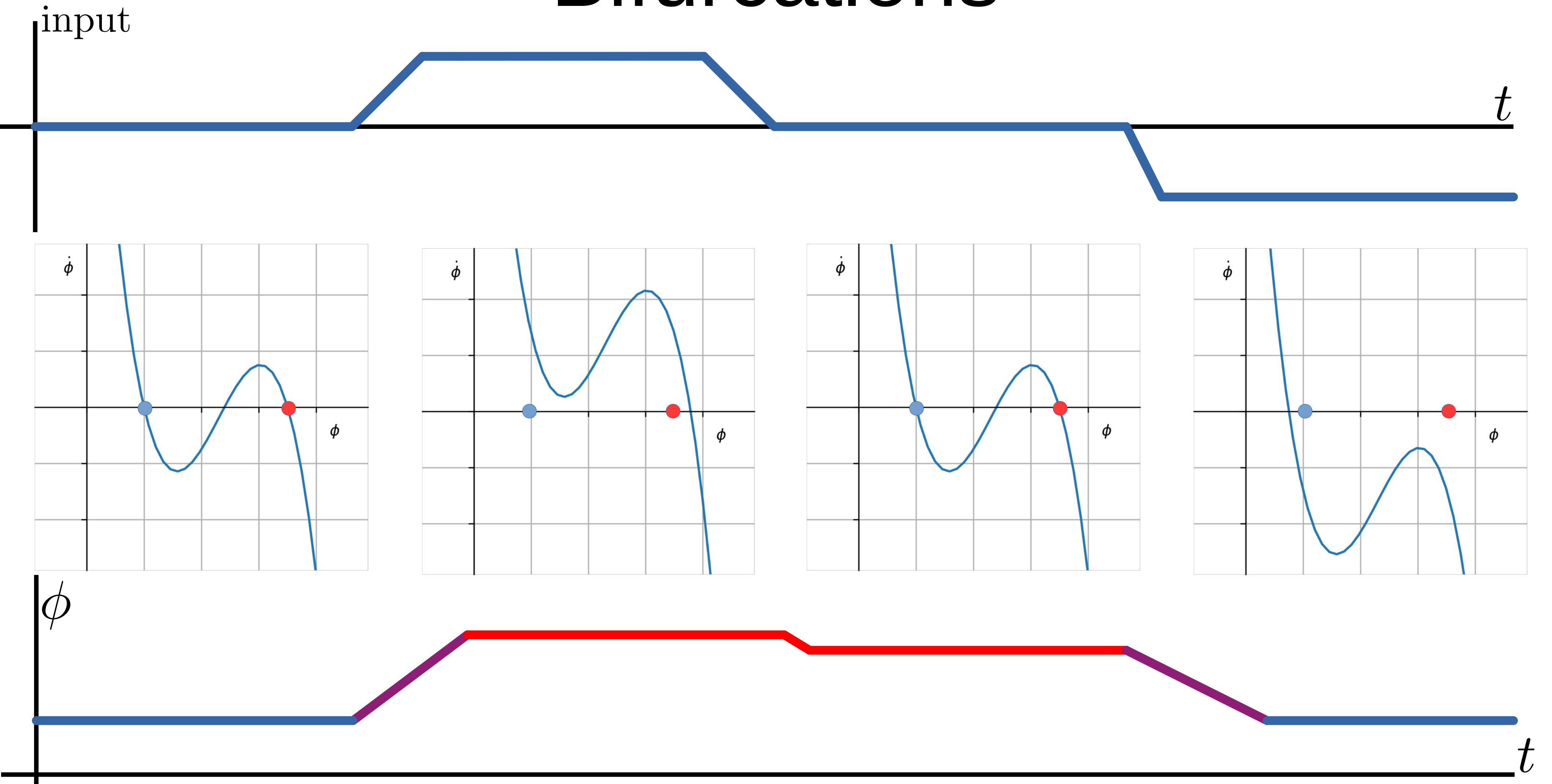
Bifurcations

Fixed points appear and vanish

- ➔ Continuous change in a parameter causes discrete change of behavior



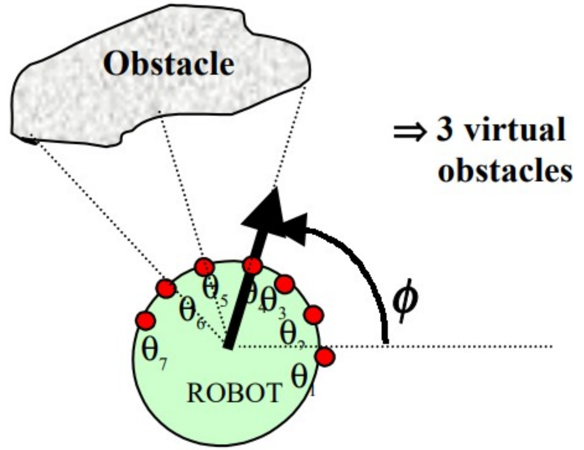
Bifurcations

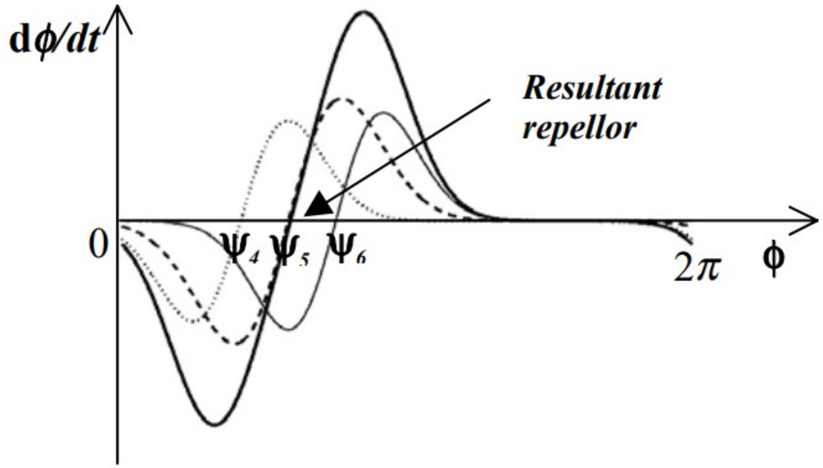


Symbolic & sub-symbolic

- So far we need know exactly where obstacles are and have one angle per obstacle
- But how to do this in a mobile robot in an unknown environment?

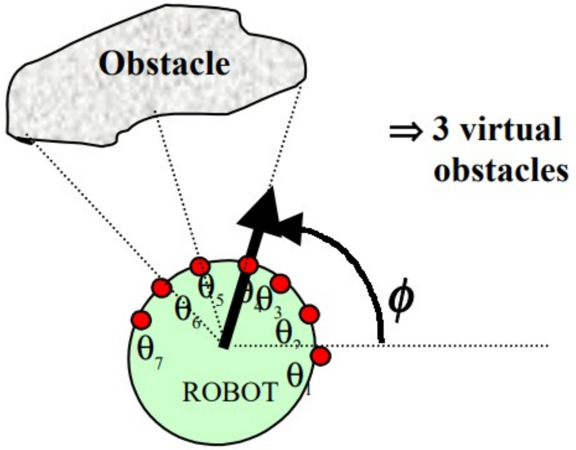
➔ Set up several forcelets that overlap and build an attractor at the appropriate point, ideally have a bifurcation when enough space between obstacles

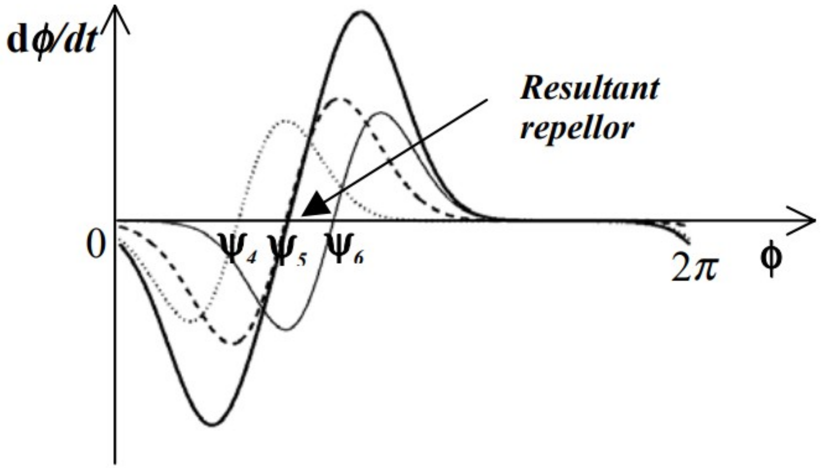




Symbolic & sub-symbolic

- One forecelet per sensor
- Each votes to turn away from its respective direction
- Strength of vote modulated by sensor value





Simulations

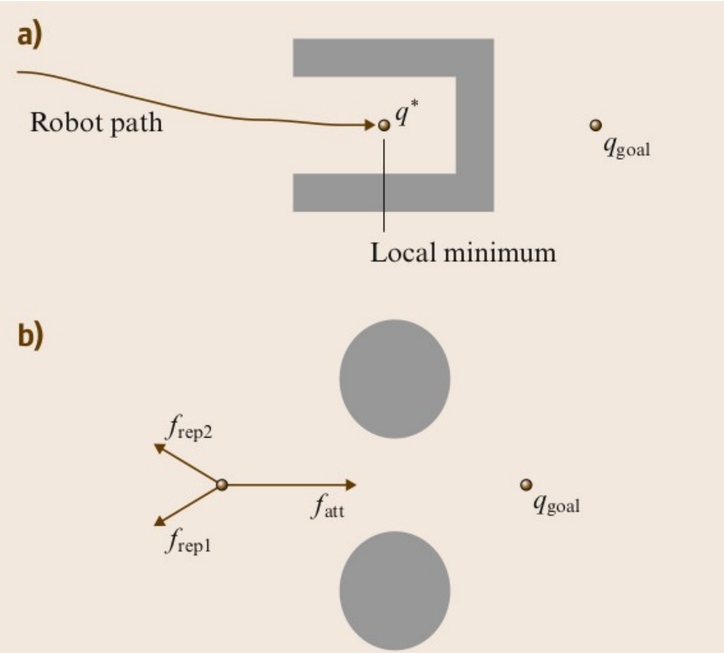
- Pin robot, demonstrate
  - Forcelet buildup when obstacle approaches
  - Two obstacles and a bifurcation
- Unpin rotation
  - Target attraction
  - Obstacle avoidance
- Unpin linear speed
  - See overall behavior

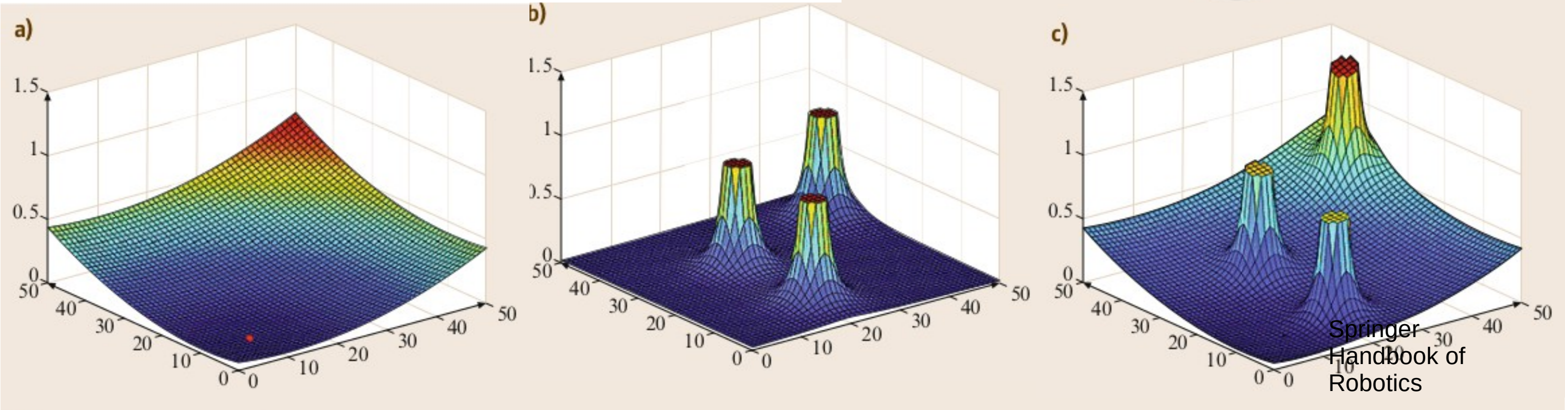
Other approaches to planning

- Potential fields
- Vector field histogram
- Cell decomposition
- Rapid random trees
- Roadmaps
- Behavior based robots

Potential fields

- Idea from physics
- Roll down the potential hill
- Notoriously vulnerable to local minima
- Very well researched, many fixes

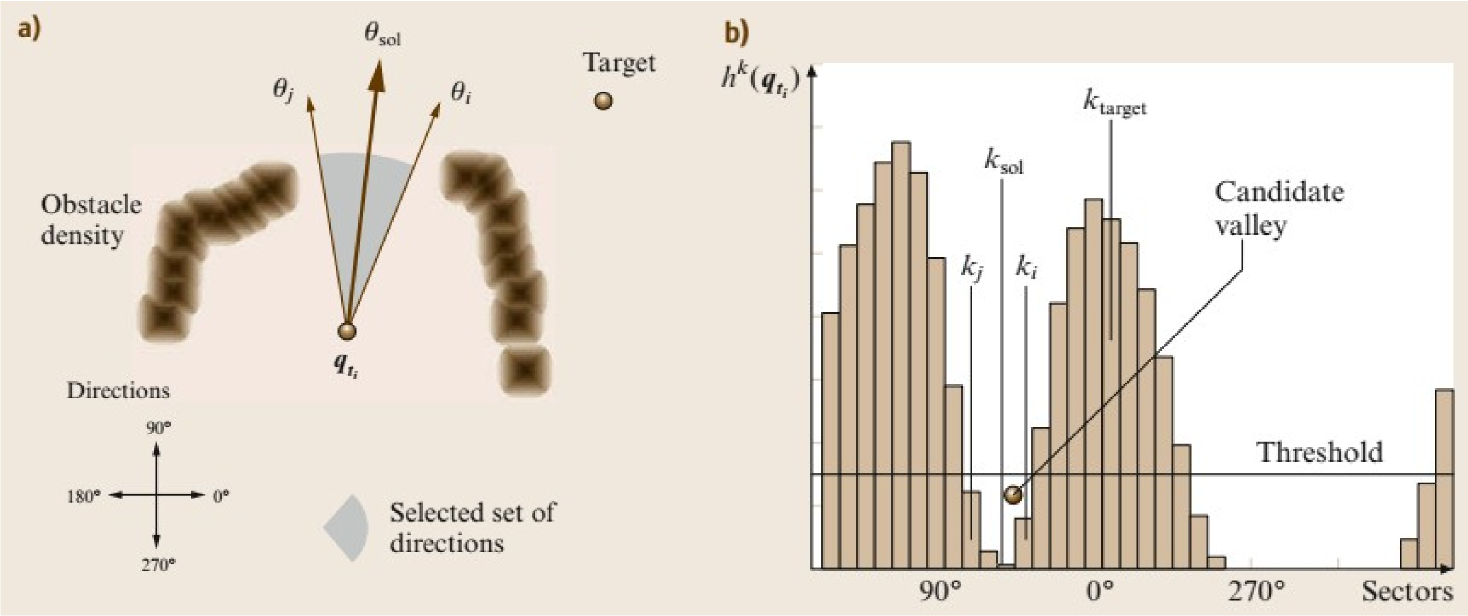




Springer Handbook of Robotics

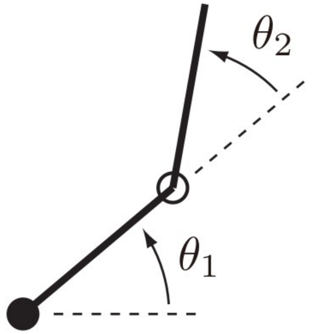
Vector field histogram

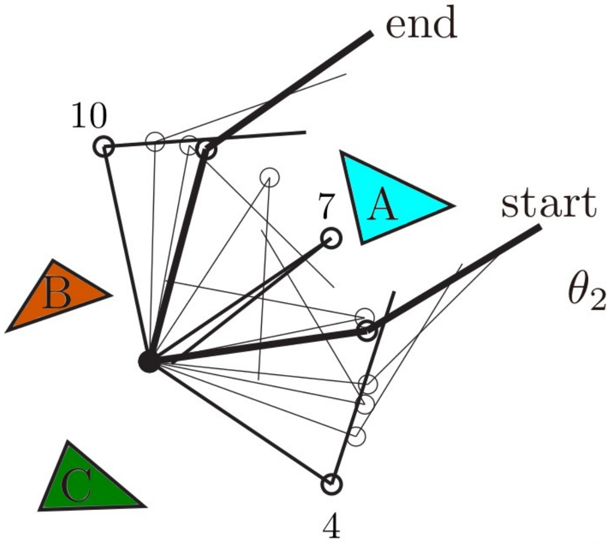
- First build distribution of probable obstacle positions
- Then pick a valley close to target

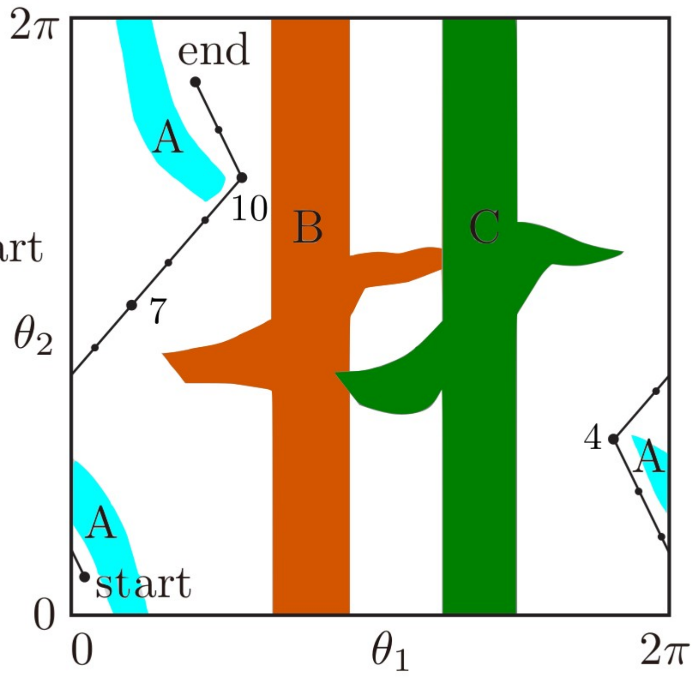


Springer Handbook of Robotics

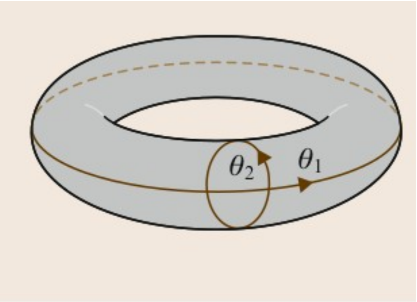
Configuration space







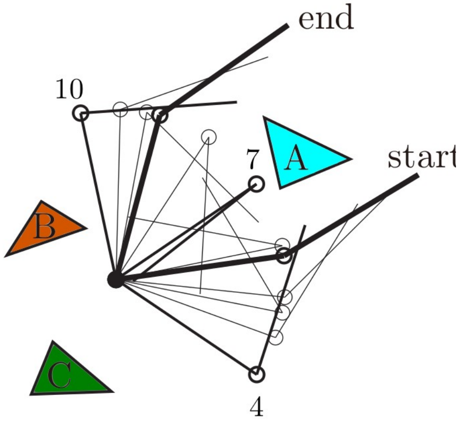
Park, Lynch –
Modern robotics

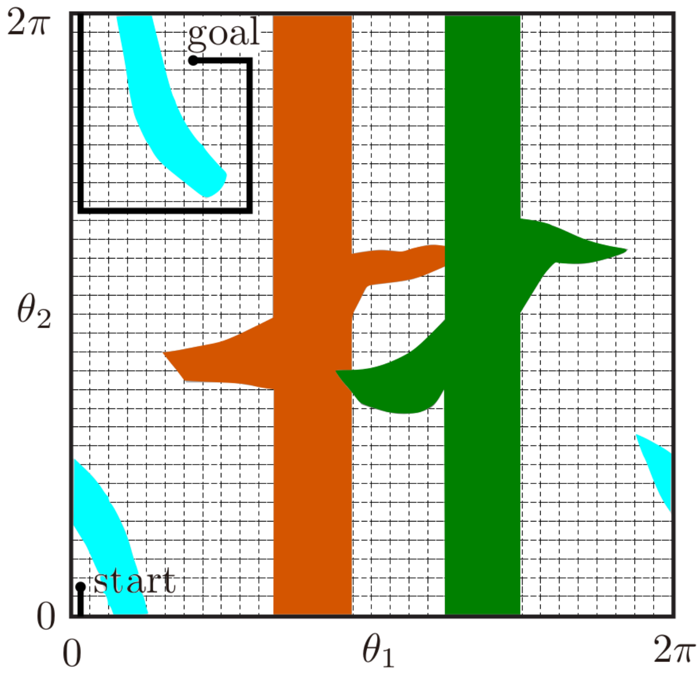


Springer
Handbook of
Robotics

Cell decomposition

- Take a map of the entire configuration space
- Graph search
- Scales terribly in higher dimensions



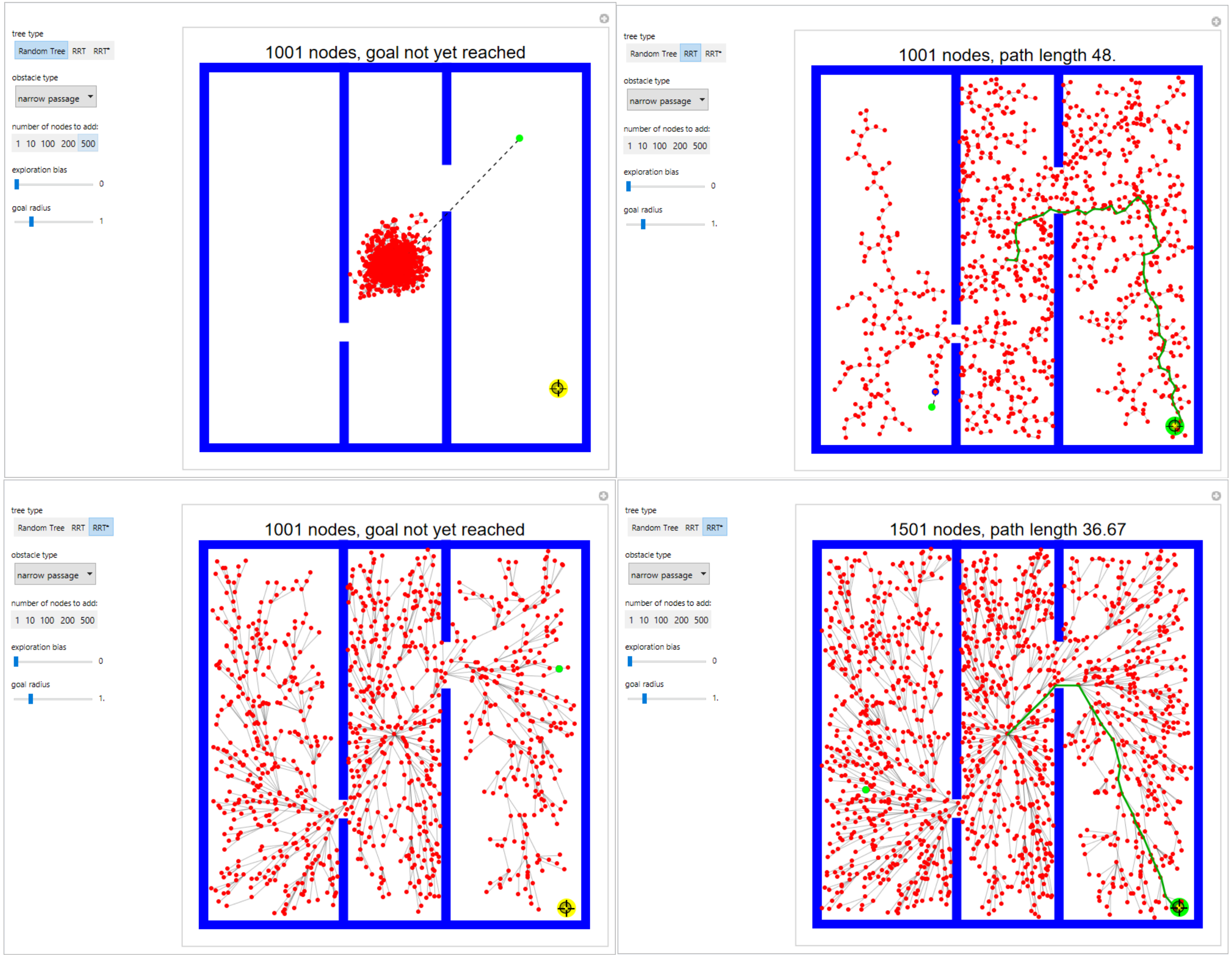


Park, Lynch – Modern robotics

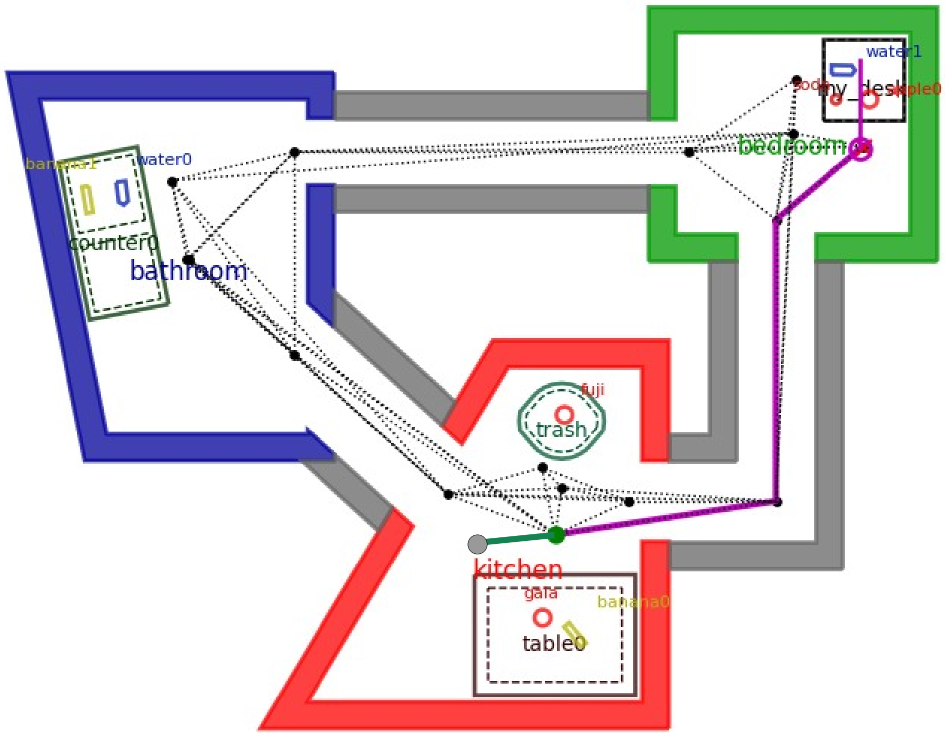
Rapid random trees

Initialize empty tree T

- Draw samples from space
- Identify closest node in graph
- Draw line from node to point, up to max distance, attach as limb to tree
- If in goal region, break


tree type
Random Tree RRT RRT*
obstacle type
narrow passage
number of nodes to add:
1 10 100 200 500
exploration bias
0
goal radius
1
1001 nodes, goal not yet reached
tree type
Random Tree RRT RRT*
obstacle type
narrow passage
number of nodes to add:
1 10 100 200 500
exploration bias
0
goal radius
1.
1001 nodes, path length 48.
tree type
Random Tree RRT RRT*
obstacle type
narrow passage
number of nodes to add:
1 10 100 200 500
exploration bias
0
goal radius
1.
1001 nodes, goal not yet reached
tree type
Random Tree RRT RRT*
obstacle type
narrow passage
number of nodes to add:
1 10 100 200 500
exploration bias
0
goal radius
1.
1501 nodes, path length 36.67

Roadmaps



pyrobosim

- Sample a part of the world for “highways”
- Use local planner to get to and from highways

Behavior

- Take various fairly simple modes of behavior
- Dynamically switch between them
- E.g.: A mouse is exploring. Then it finds food and eats. Then it sees a cat and flees.