Robot Navigation with a Polar Neural Map

Michail G. Lagoudakis

Department of Computer Science

Duke University

Anthony S. Maida

Center for Advanced Computer Studies

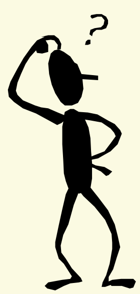
University of Southwestern Louisiana

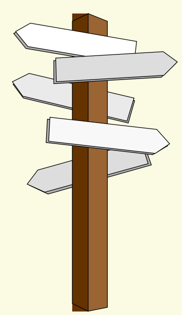
16th National Conference on Artificial Intelligence (AAAI-99)

July 18-22, 1999 - Orlando, Florida

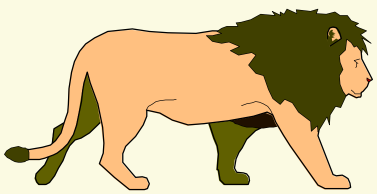
Mobile Robot Navigation

- ***Global Navigation***
  - Map-Based
  - Deliberative
  - Slow





- ***Local Navigation***
  - Sensory-Based
  - Reactive
  - Fast



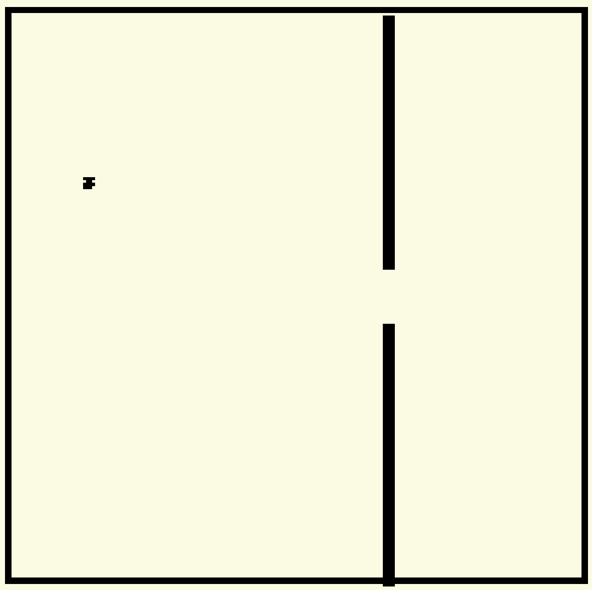
Global Path Planning Methods

- ✔ ***Distance Transform***
  - (Jarvis, 1993)
  - Fast
  - Non-Smooth Paths

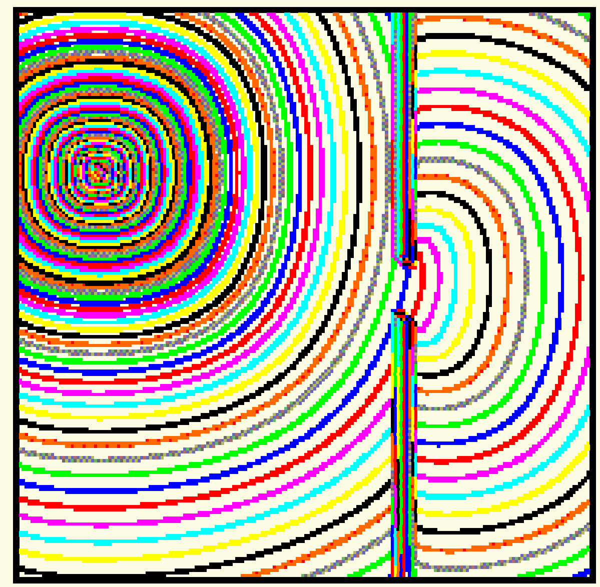
- ✔ ***Harmonic Functions***
  - (Connoly et al., 1990)
  - Slow
  - Smooth Paths

- ✔ ***Neural Maps***
  - (Glasius et al., 1995)
  - Quite Fast
  - Smooth Paths

Main Idea (1)

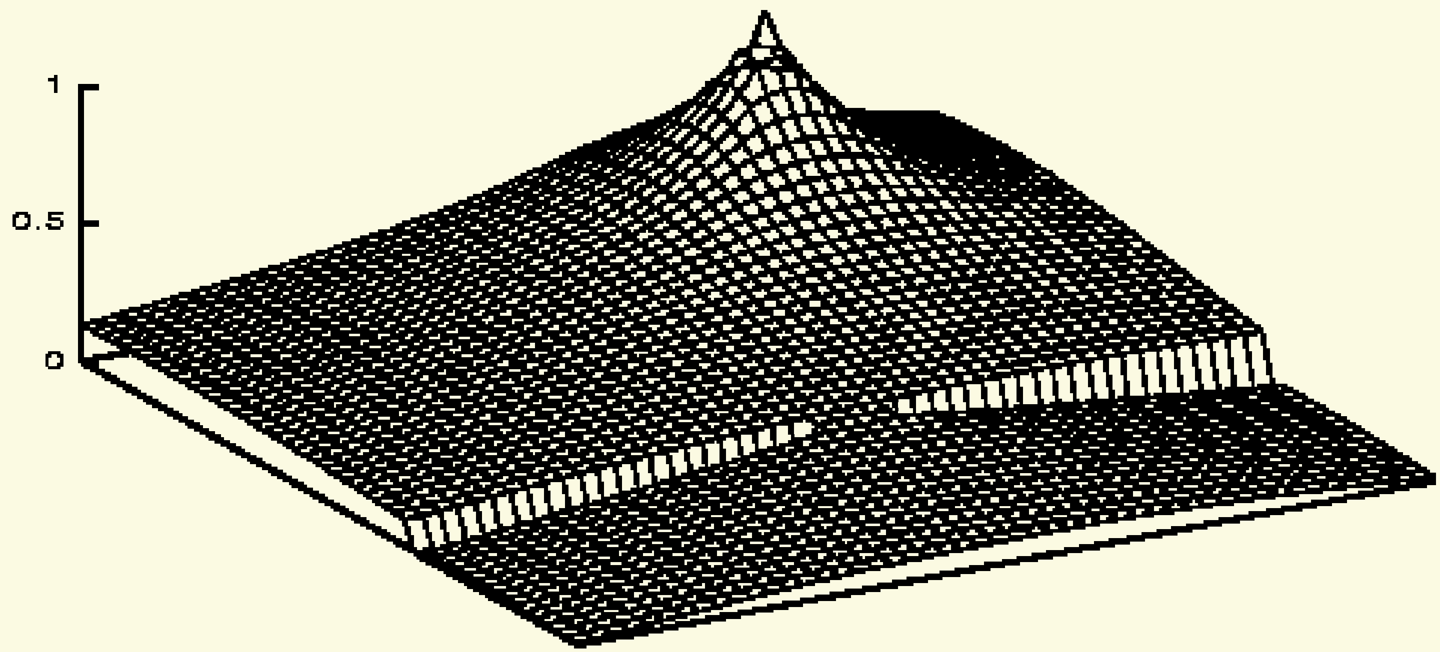


Create a model of the robot's environment.



Simulate diffusion from the target position.

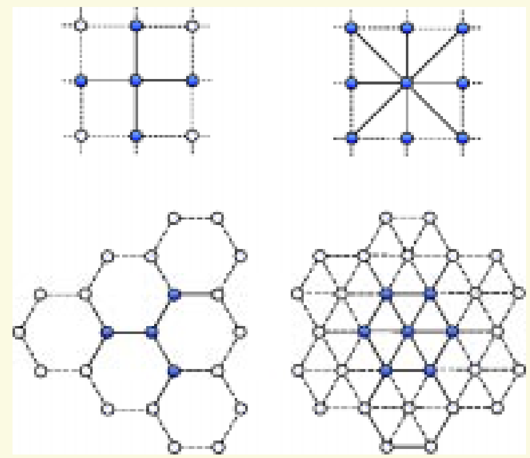
Main Idea (2)



Find a path from any initial position to the target by steepest ascent (maximum gradient following) on the navigation landscape.

Neural Maps for Path Planning

- ✔ A ***neural map*** is "*a localized neural representation of signals in the outer world*" [Amari, 1989]
- ✔ The map is a discrete topologically ordered representation of the robot's configuration space.
- ✔ Information on the map:
  - – ***Target*** configuration(s)/unit(s)
  - – ***Obstructed*** configurations/units
- ✔ The ***weight*** between two units i and j reflects the ***cost of moving*** between the corresponding configurations c_j and c_j.



Sample uniform unit topologies and connectivity

Neural Map Diffusion Dynamics

- ✔ External (Sensory/Map) Input

$$\theta_i(t) = \begin{cases} +\infty & i \text{ is target at time } t \\ -\infty & i \text{ is obstacle at time } t \\ 0 & \text{otherwise} \end{cases}$$

- ✔ Lateral Connections

  $\rho(i, j) = \text{Euclidean Distance}\,(i, j)$

  $r = \text{range of connections}$

$$w_{ij} = \begin{cases} 0 & \rho(i, j) = 0 \\ \frac{1}{\rho(i,j)} & 0 < \rho(i, j) \le r \\ 0 & r < \rho(i, j) \end{cases}$$

- ✔ Nonlinear Activation Function

$$\Phi_\beta(x) = \begin{cases} 0 & x \le 0 \\ \tanh(\beta x) & x > 0 \end{cases}$$

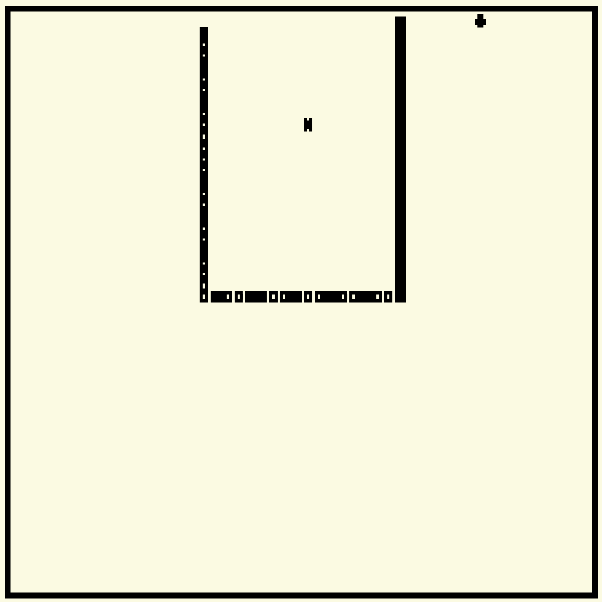
- ✔ Activation Update Equation

$$v_i(t+1) = \Phi\Big(\sum_j w_{ij} v_j(t) + \theta_i(t)\Big)$$

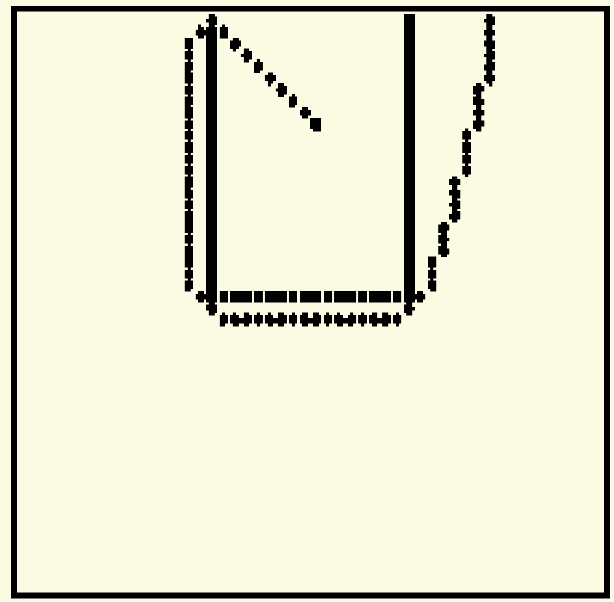
- ✔ Equilibrium State

$$v_i(t+1) = v_i(t)$$

Path Planning Example 1

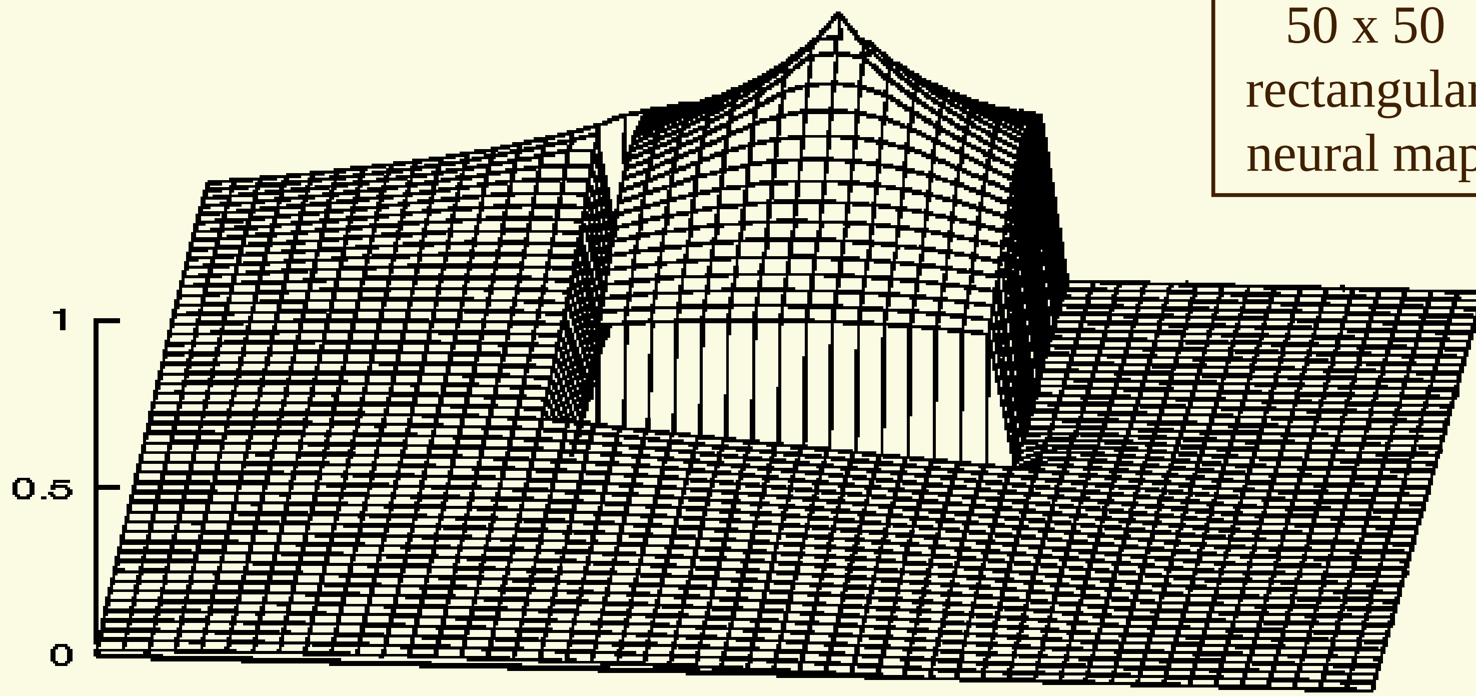


Target (middle) and initial position (up right).



Obstacle-free path from initial position to the target.

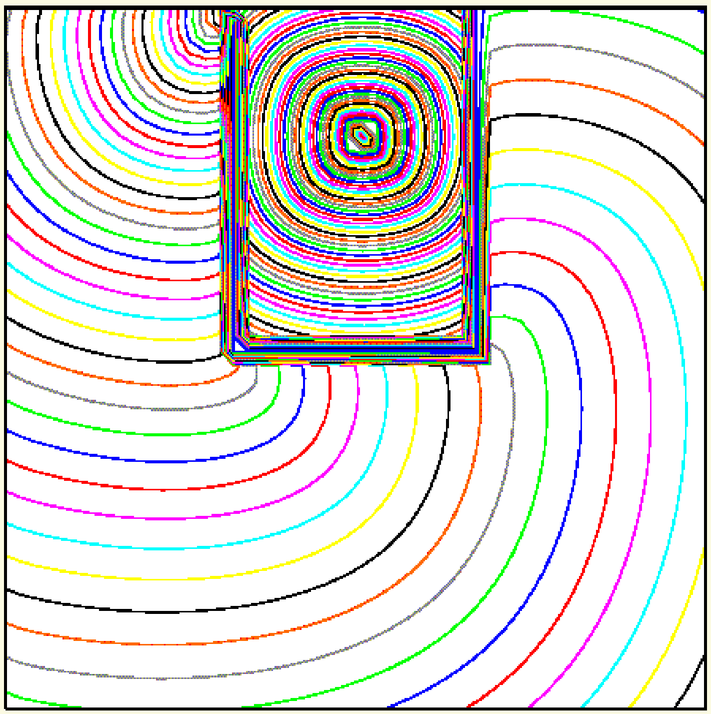
Path Planning Example 1

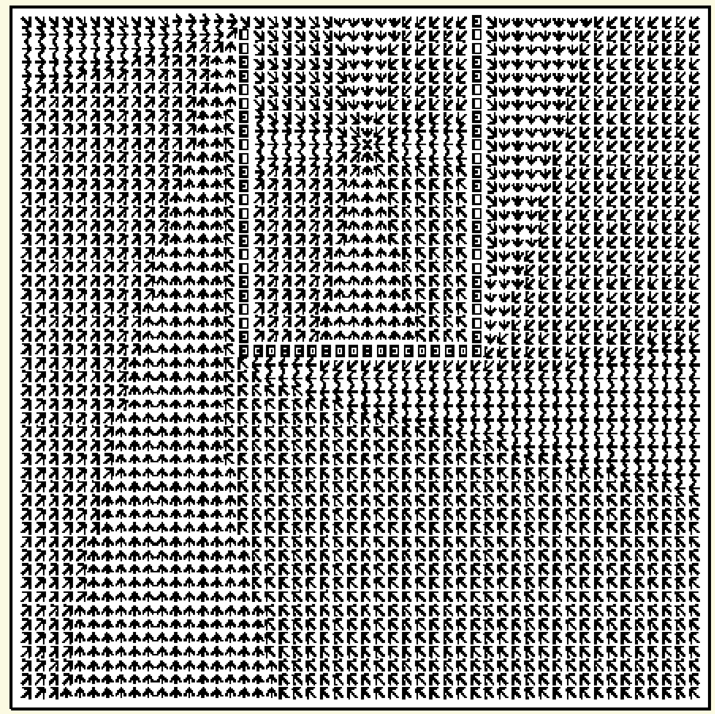


50 x 50 rectangular neural map

Activation landscape formed on the neural map at equilibrium.

Path Planning Example 1

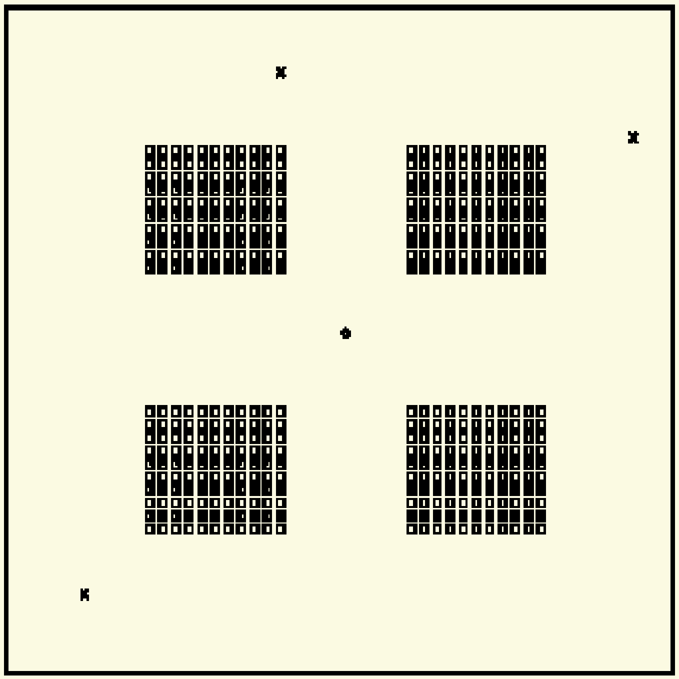


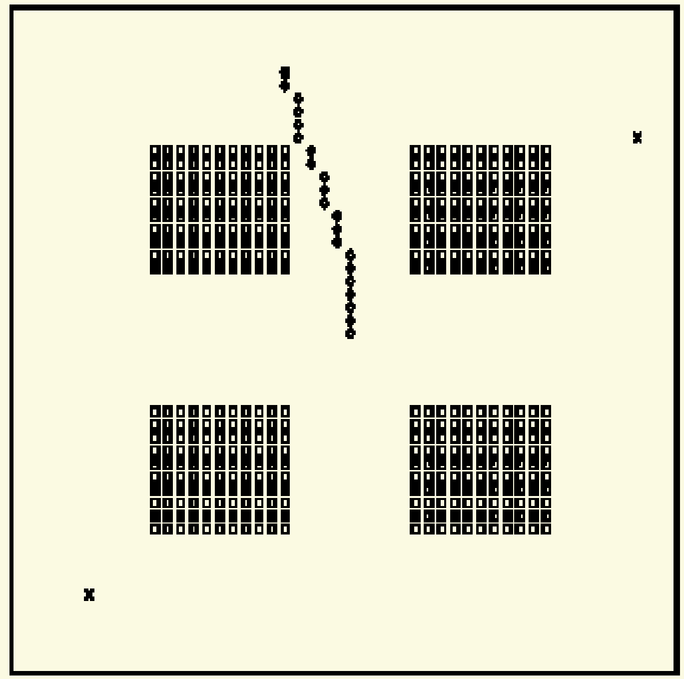


Activation diffusion on the neural map.

Navigation map for the given target.

Path Planning Example 2



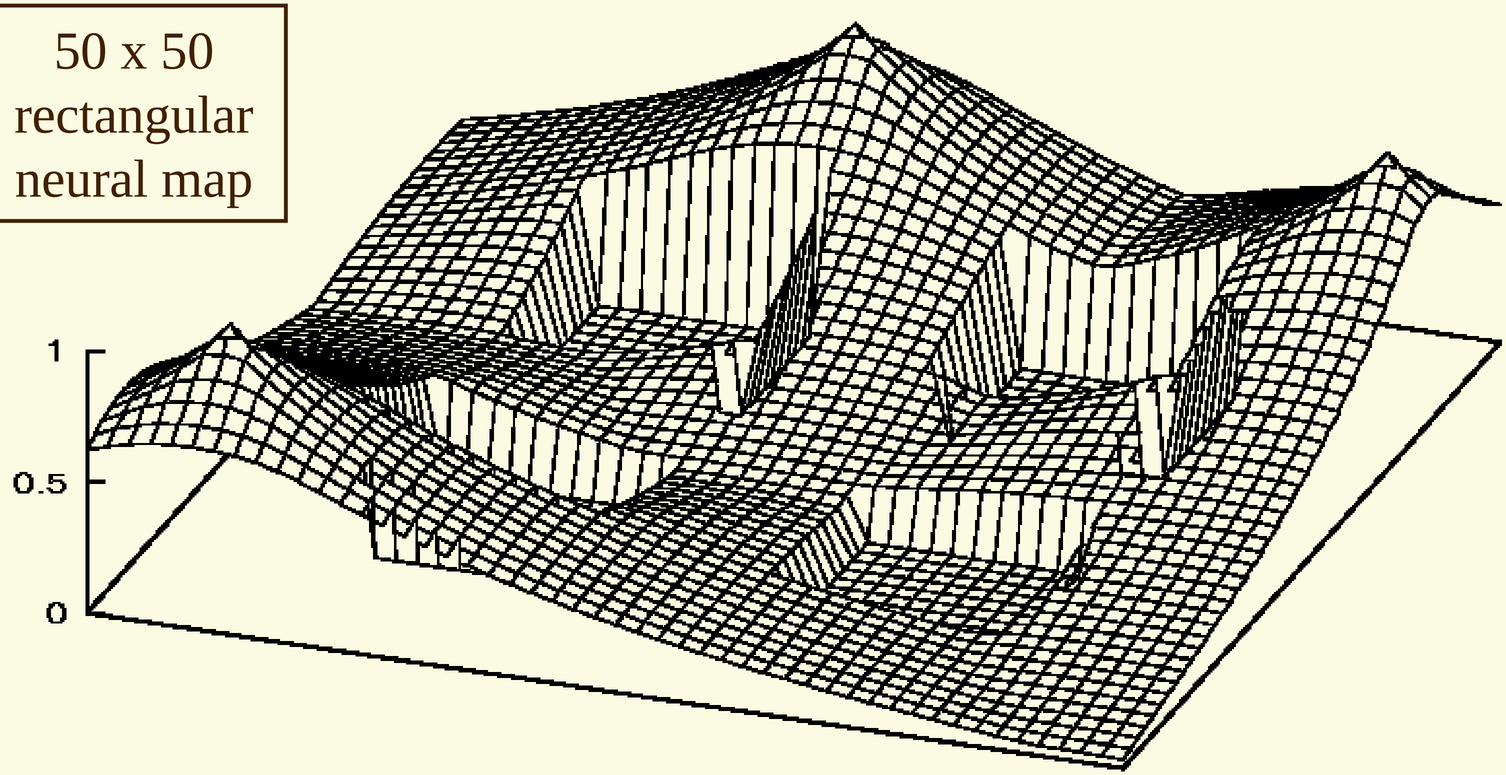


Initial position (middle) and three targets.

Obstacle-free path to the closest target.

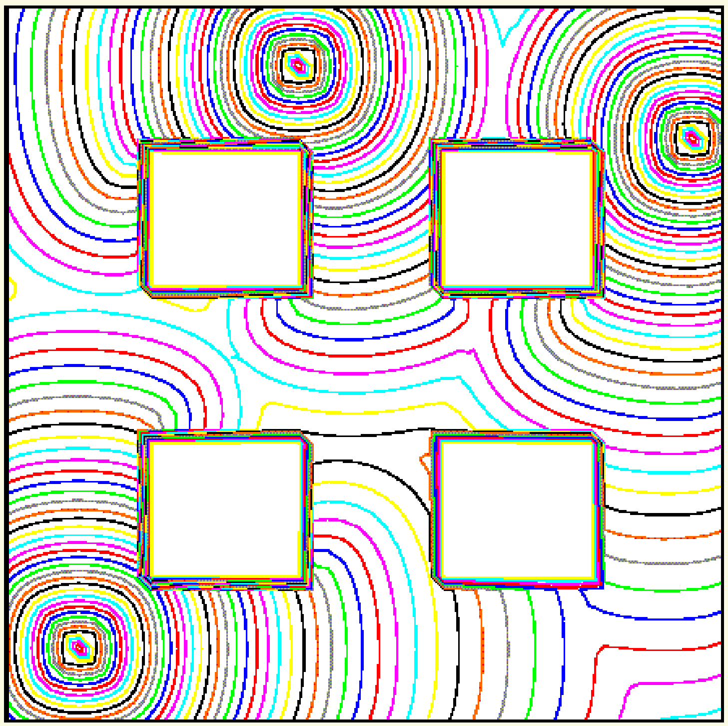
Path Planning Example 2

50 x 50
rectangular
neural map

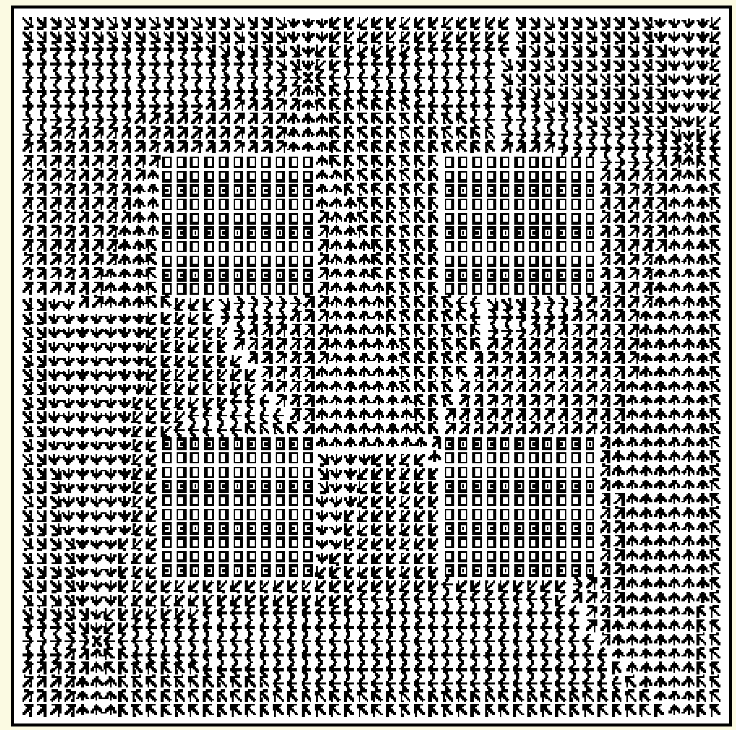


Activation landscape formed on the neural map at equilibrium.

Path Planning Example 2



Activation diffusion on the neural map.



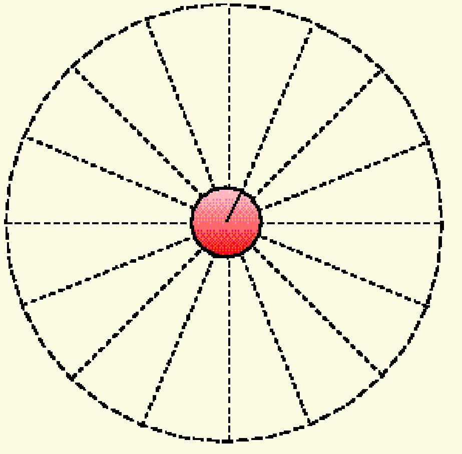
Navigation map for the given targets.

Nomad 200 Mobile Robot

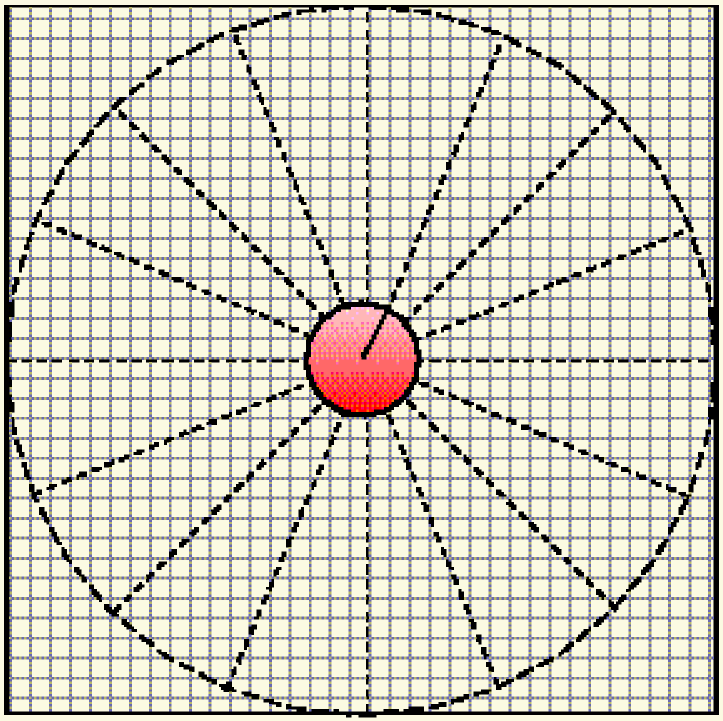
- ✔ Nonholonomic Mobile Base
- ✔ Zero Gyro-Radius
- ✔ Max Speeds: 24 in/sec, 60 deg/sec
- ✔ Diameter: 21 in, Height: 31 in
- ✔ Pentium-Based Master PC
- ✔ Linux Operating System
- ✔ Full Wireless 1.6 Mbps Ethernet
- ✔ 16 Sonar Ring (6 in - 255 in)
- ✔ 20 Bump Sensors

Neural Maps for Local Navigation

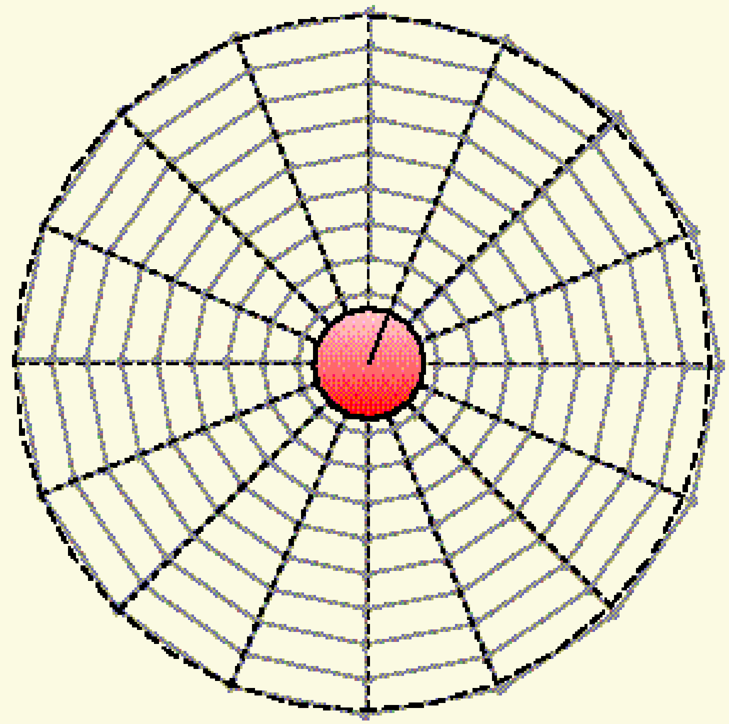
- ✔ No global/map information!
- ✔ Sensory information
  - – Egocentric view
  - – Circular range
  - – Decaying resolution
- ✔ A neural map **can be used** if adapted appropriately to account for the sensory and motor capabilities of the robot!



“Bad” and “Good” Organization



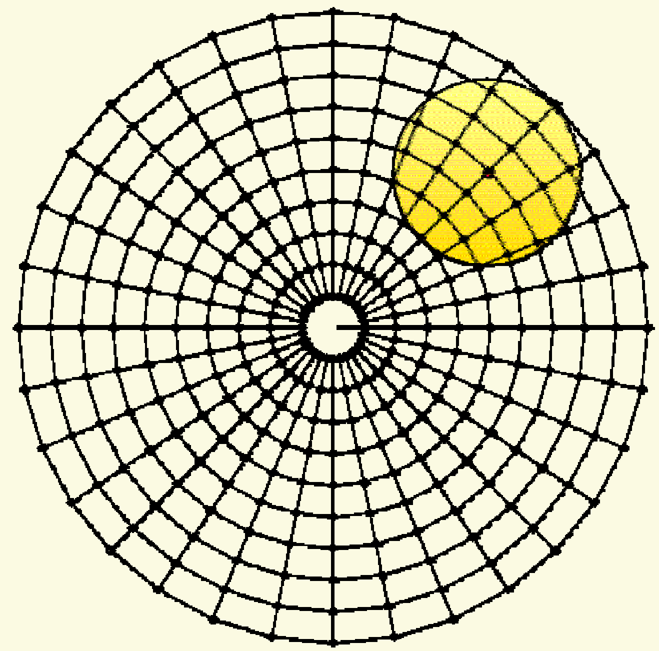
Rectangular Topology



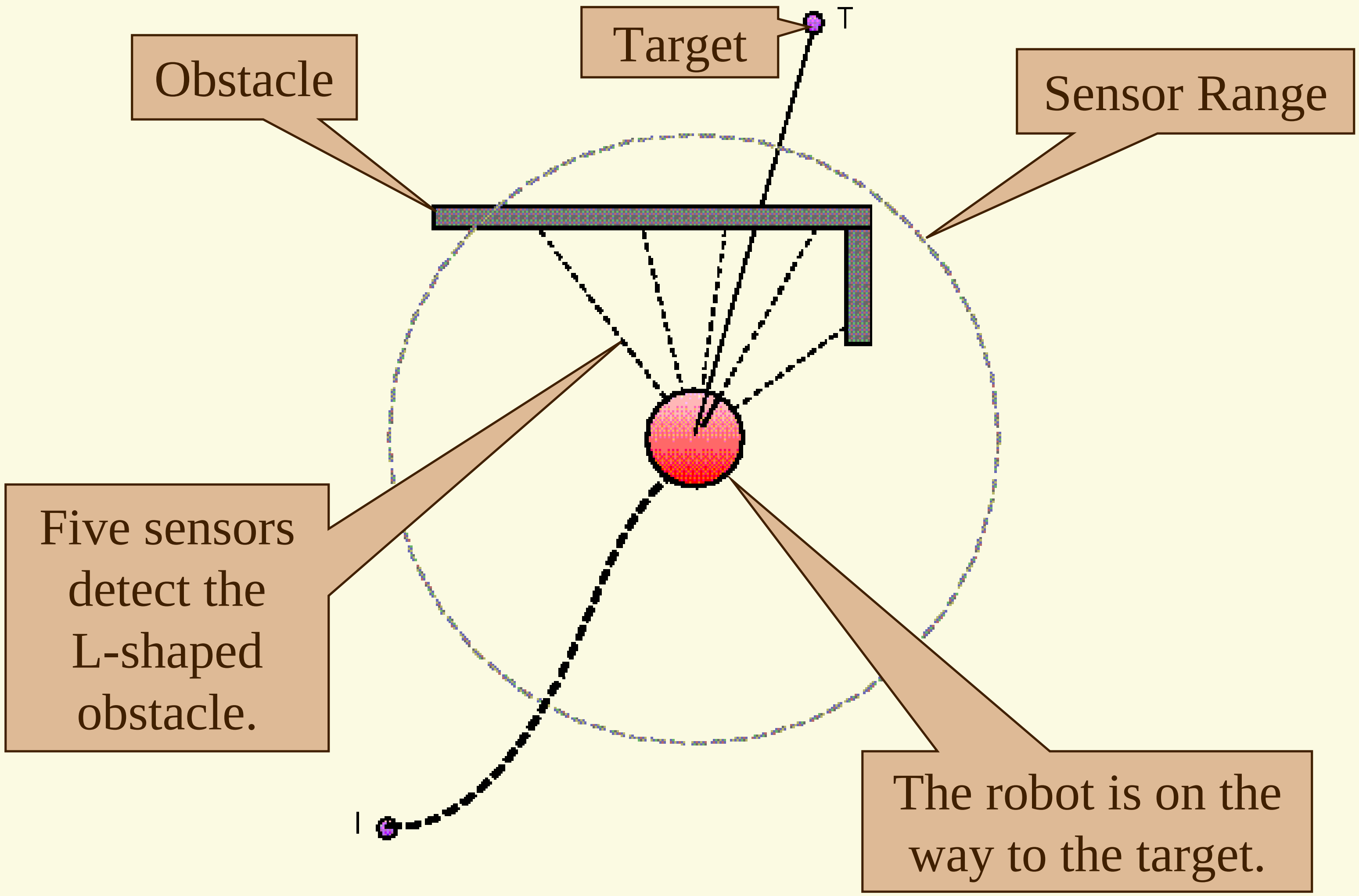
Polar Topology

The Polar Neural Map

- ✔ Represents the local space.
- ✔ Resembles the distribution of sensory data.
- ✔ Provides higher resolution closer to the robot.
- ✔ Conventions:
  - Inner Ring: Robot Center
  - Outer Ring: Target Direction
- ✔ Robot's "*Working Memory*"

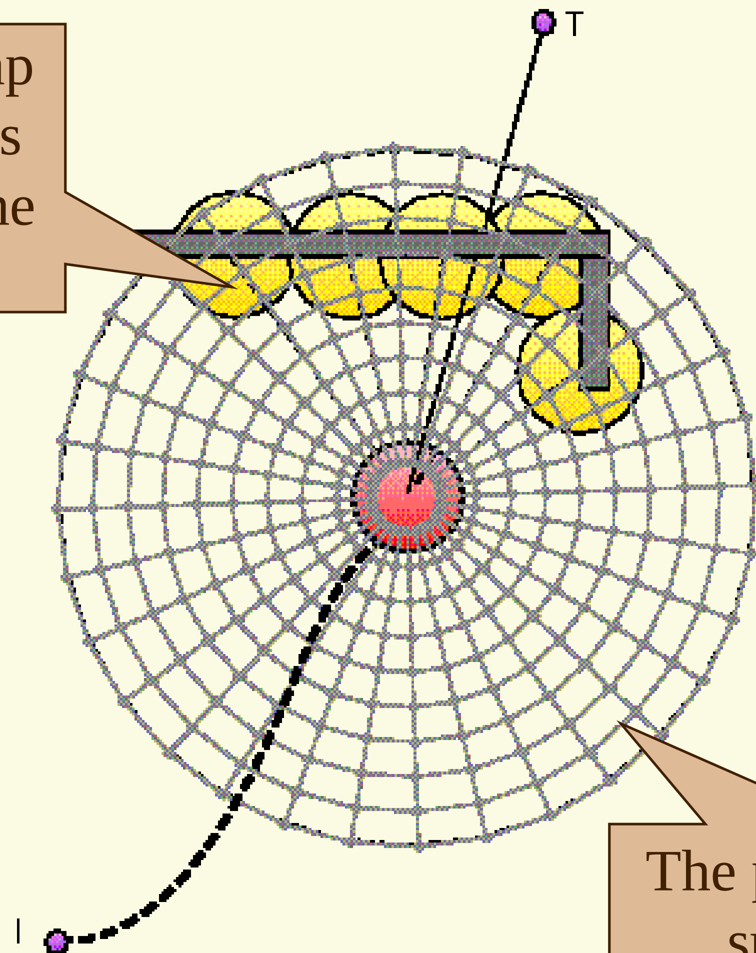


Incremental Path Planning (1)



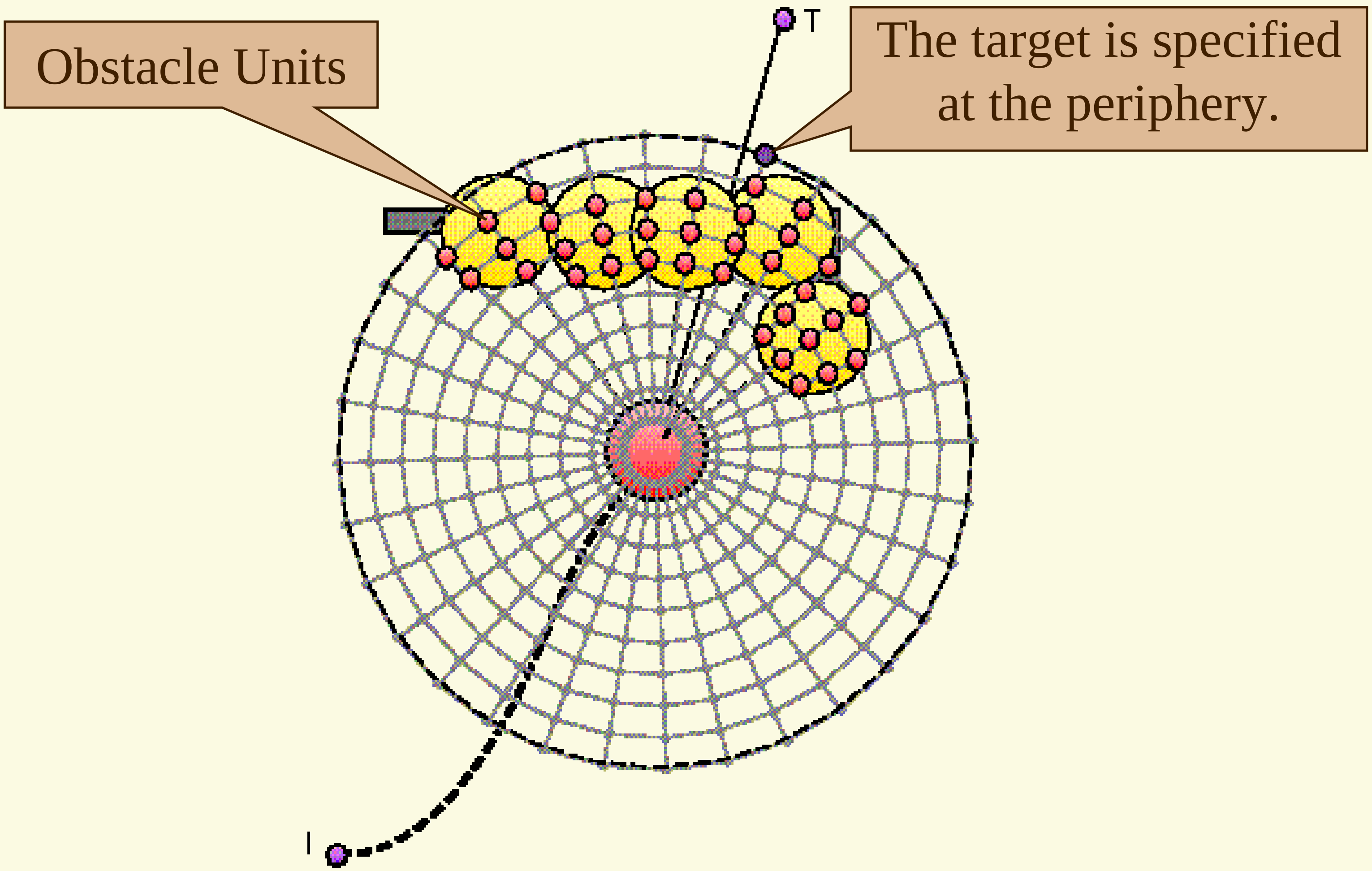
Incremental Path Planning (2)

Areas of the map characterized as obstructed by the sensor data.

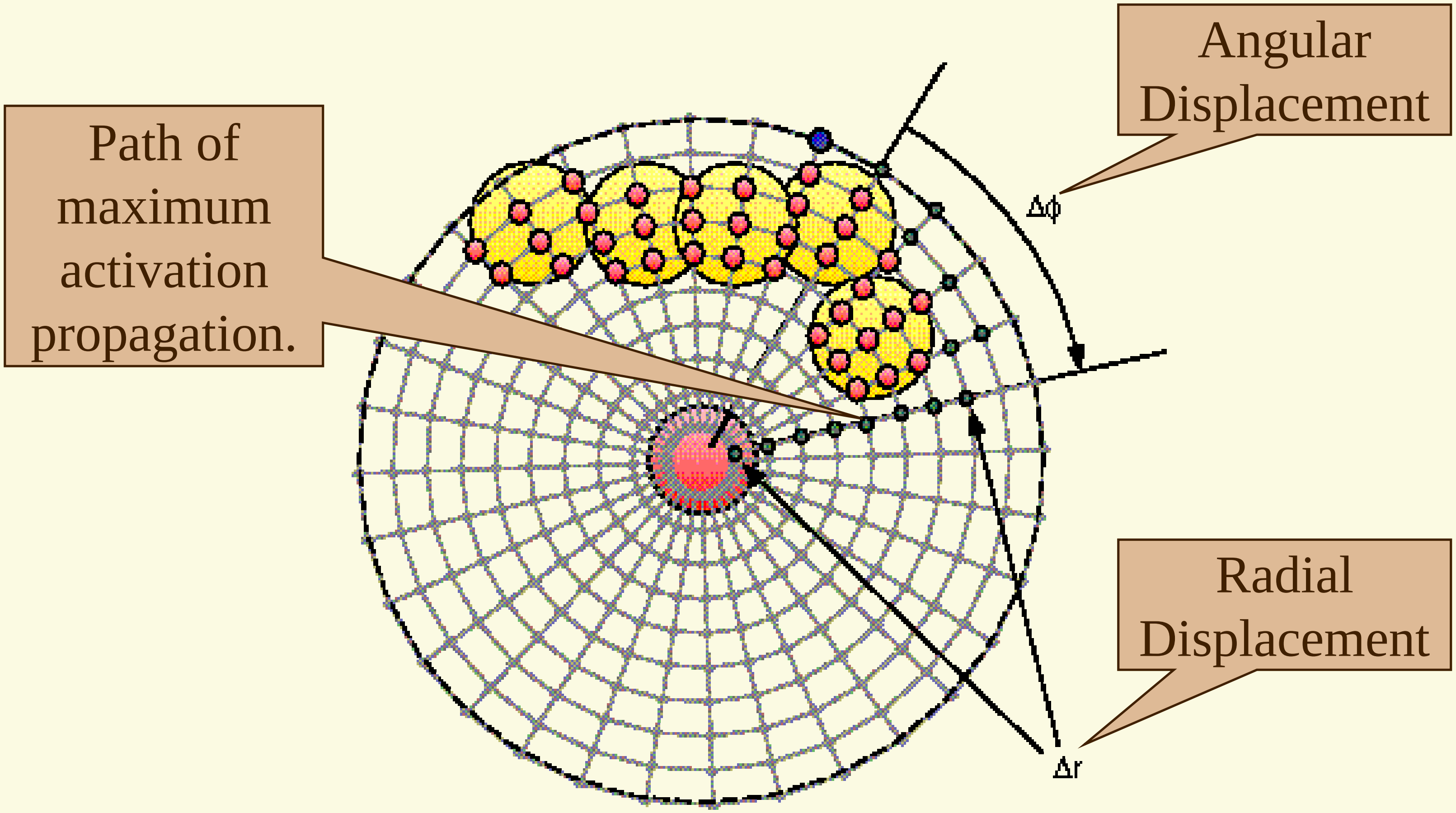


The polar neural map superimposed.

Incremental Path Planning (3)



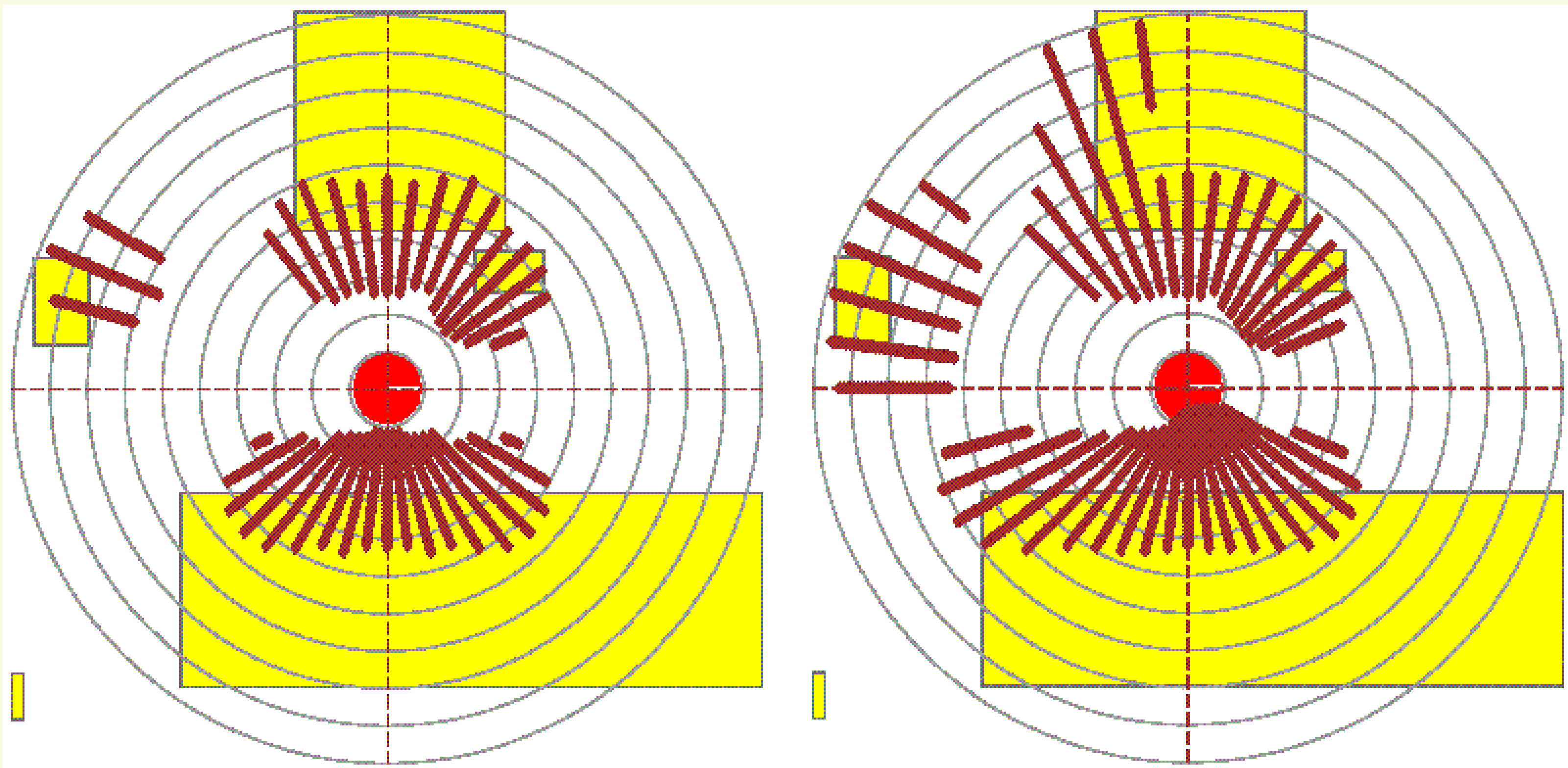
Incremental Path Planning (4)



Sonar Short-Term Memory

- ✔ Maintain a *window* of the last *n* sonar scans
  - – corresponding to about 2-3 seconds of real time
- ✔ Project all data to the current position (reuse)
  - – use odometric information (locally accurate)
- ✔ Conservative View
  - – Assume that all data are correct
  - – Discard only those that fall:
    - within the physical area of the robot
    - outside the polar map

Representation on the Polar Map



100×48 Polar Map
Memory Window Size = 1

100×48 Polar Map
Memory Window Size = 10

Configuration Prediction

- **Problem:**
  - The ***action*** taken at the **end** of the current step is based on the ***perception*** of the world at the **beginning** of the current step.
- **Solution:**
  - ***Measure*** dynamically the (real) time taken for each control step.
  - ***Estimate*** the robot configuration at the end of the current step, using a model of the robot kinematics (unicycle model).
  - ***Project*** all data (sonar, target) to the predicted configuration.
  - ***Determine*** the control input using the predicted/projected data.

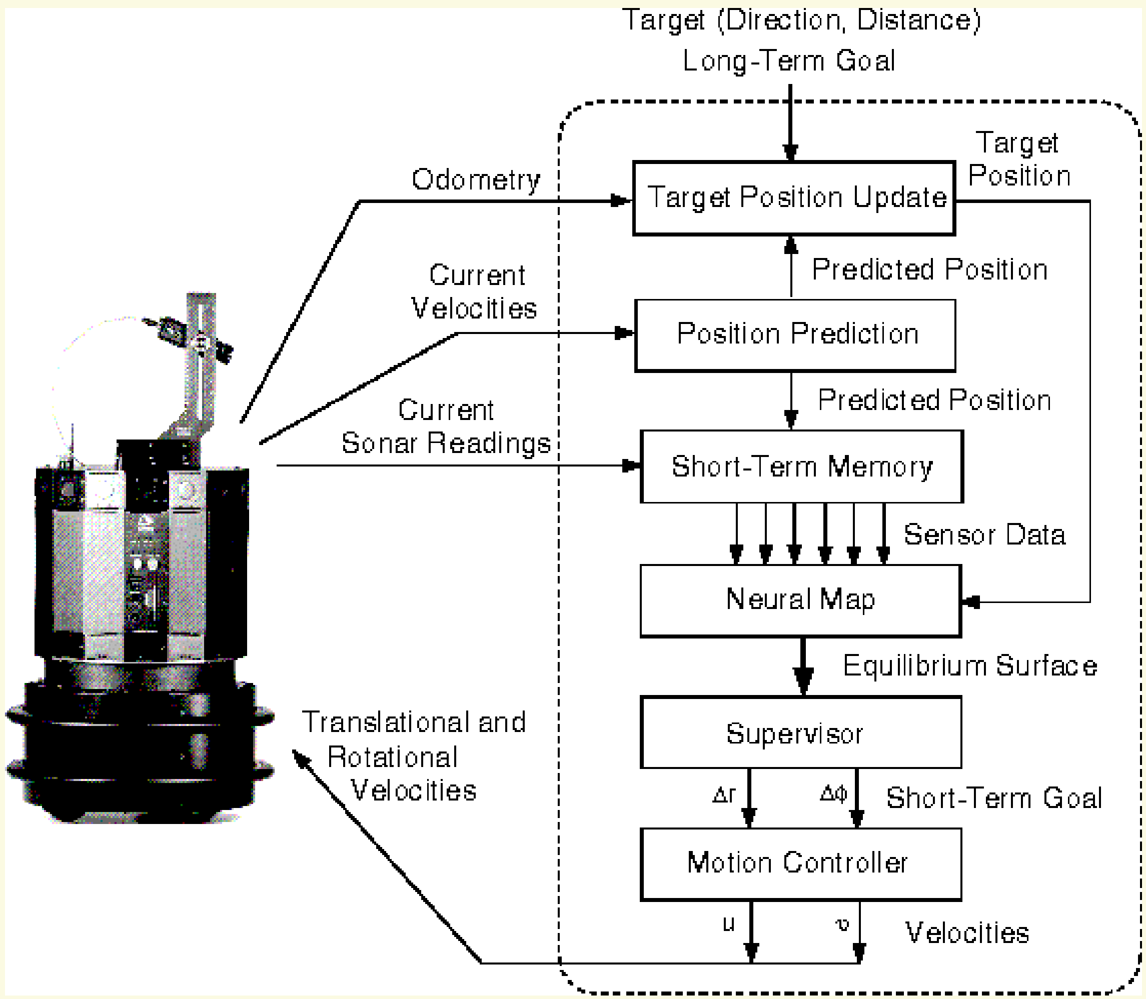
Motion Control

- ✔ Determine the control input (u,v)
  - – Translational and Rotational Velocity
- ✔ Dynamic Constraints
  - – Limited Acceleration
- ✔ Kinematic Constraints
  - – Nonholonomic System
  - – 3 degrees of freedom vs. 2 degrees of action

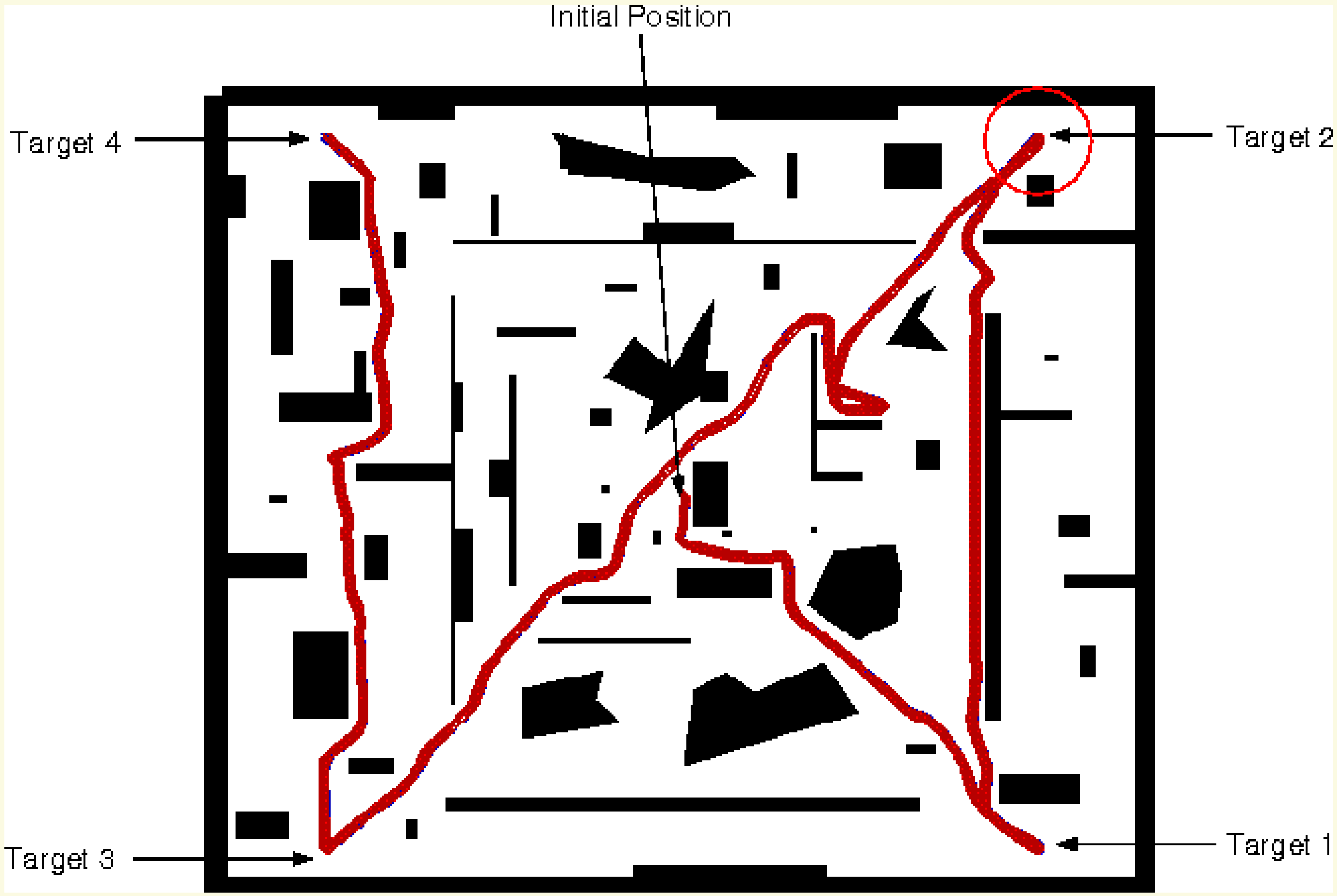
Motion Control Algorithm

- ✔ Determine the ***Dynamic Window*** (DW) [Fox et al.,1997]
- ✔ The ***Objective Function*** combines
  - – Distance error from the goal
  - – Orientation error from the goal
  - – Density of obstacles along the trajectory
- ✔ Find exhaustively the pair (u,v) that ***minimizes*** the objective function

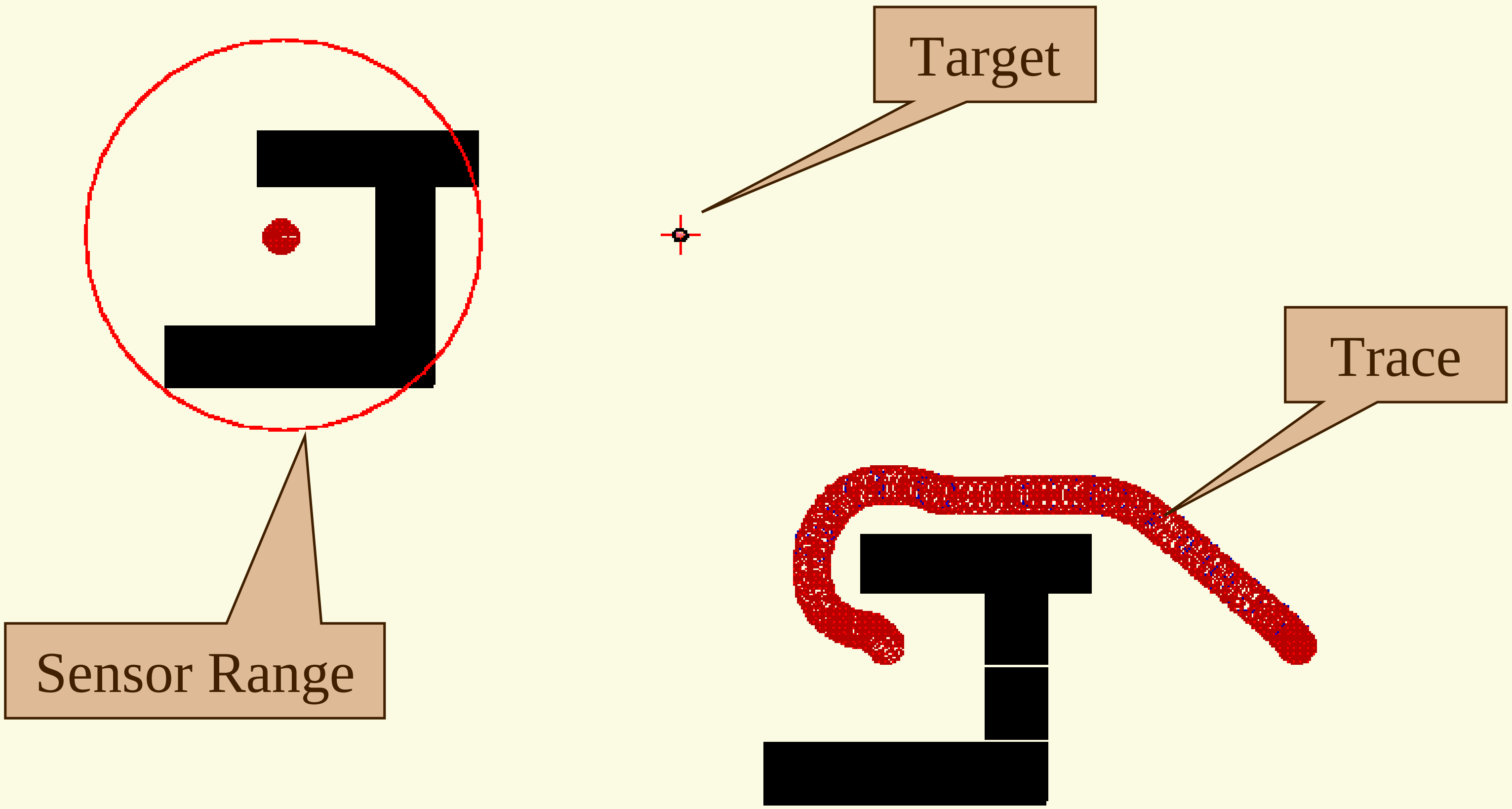
System Architecture



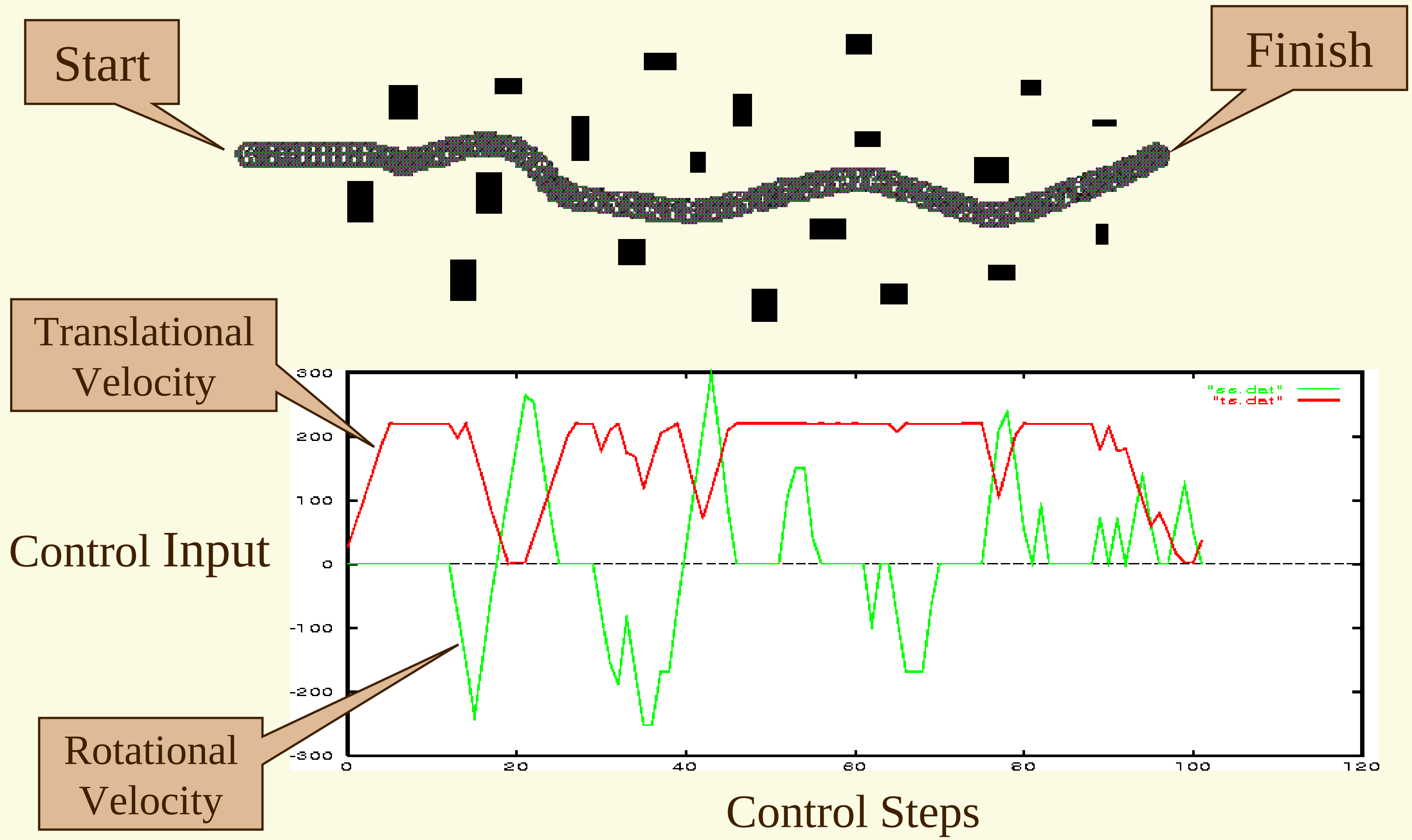
Navigation in a Simulated World



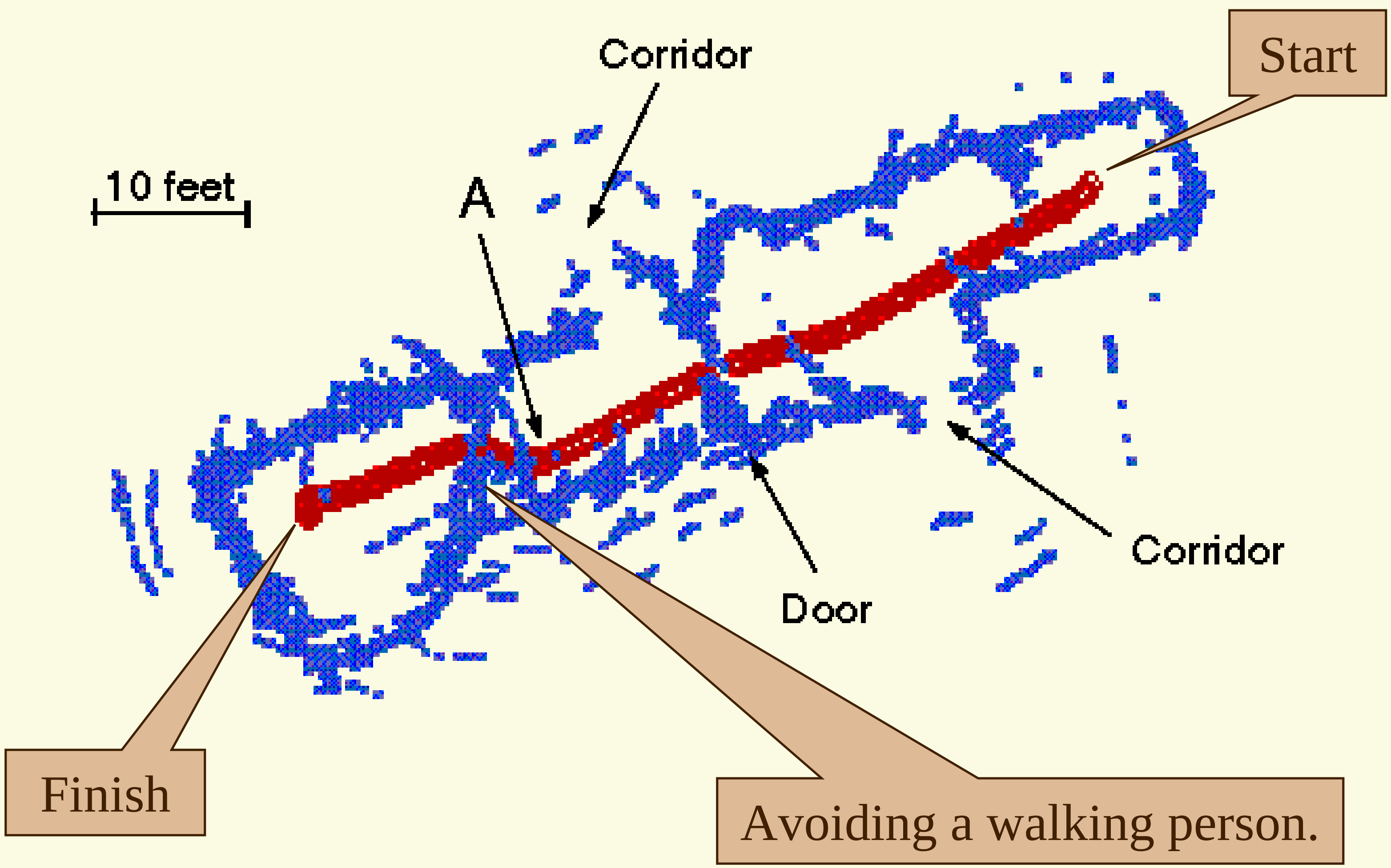
U-Shaped Obstacle



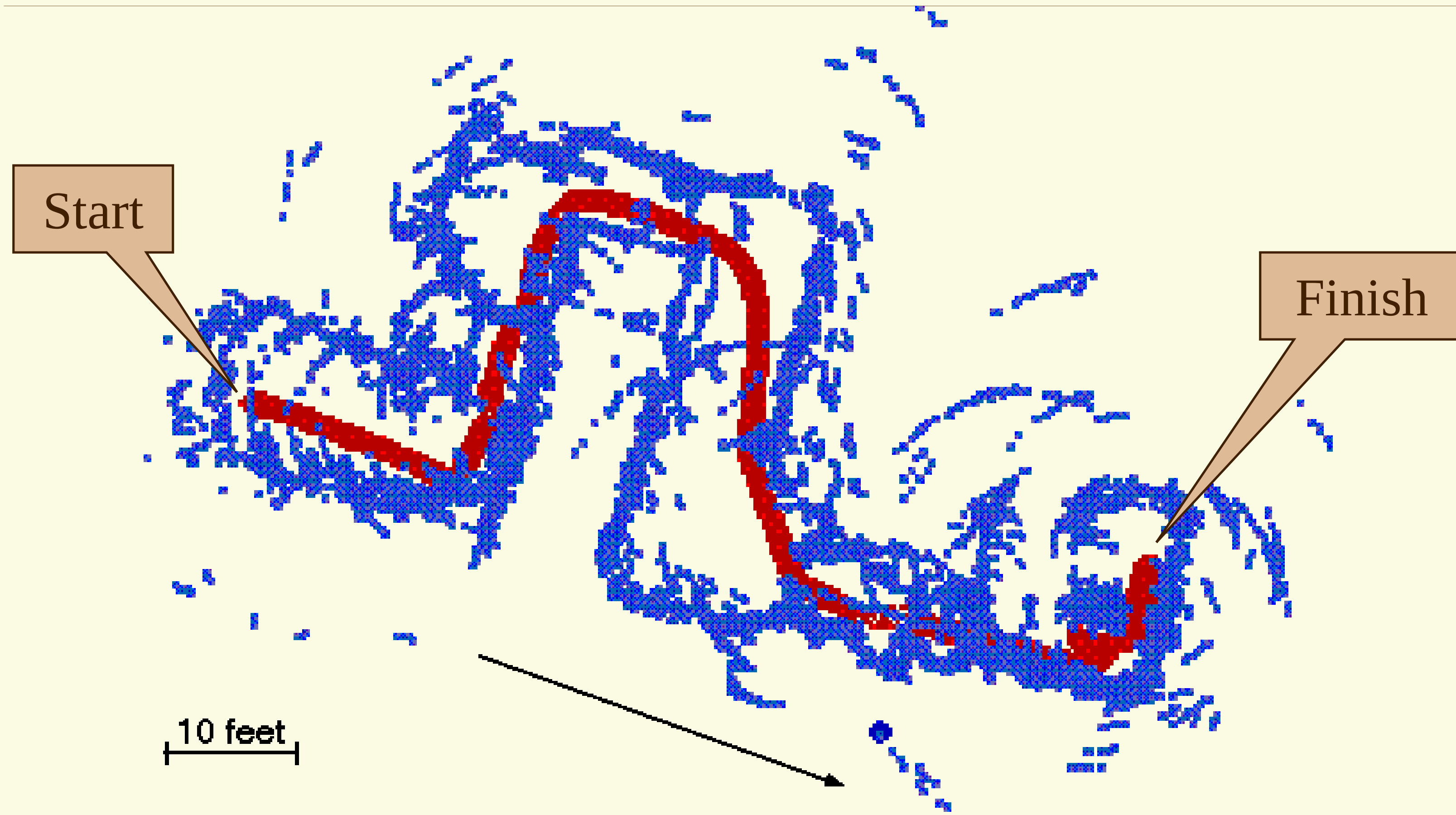
Cluttered Environment



Navigation in the Real World (1)



Navigation in the Real World (2)



The target is distant in the direction of the arrow.

Contributions

- ✔ ***The Polar Neural Map***
  - "*Working memory*" of the robot holding local (in a spatial and temporal sense) information.
- ✔ ***A complete Local Navigation System***
  - Implemented and tested on a Nomad 200 robot.

Further Information

- ✔ ***Neural Maps for Mobile Robot Navigation***
  - Lagoudakis and Maida, IEEE Intl Conf on Neural Networks, 1999.
- ✔ ***Mobile Robot Local Navigation with a Polar Neural Map***
  - M. Lagoudakis, M.Sc. Thesis, University of SW Louisiana, 1998.

Future Work

- ✔ Role of Weight Values in the Map
- ✔ Polar and Logarithmic Map
- ✔ Self-Organization of the Neural Map
- ✔ Integrated Full Navigation Method

Acknowledgments

USL Robotics and Automation Lab
Prof. Kimon P. Valavanis
Lilian-Boudouri Foundation (Greece)