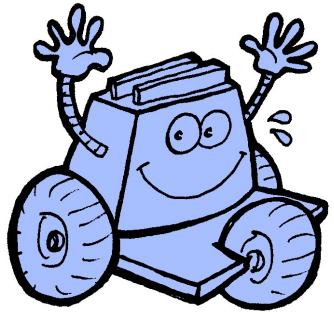
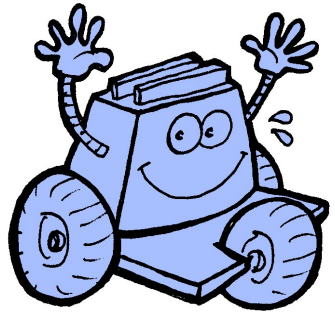
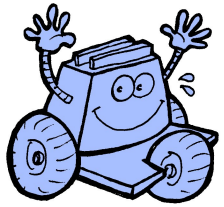


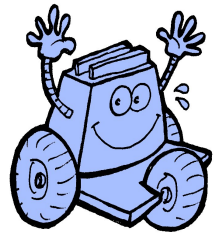


Introduction to Mobile Robots Motion Planning

Prof. Yvan Petillot

Heriot-Watt University





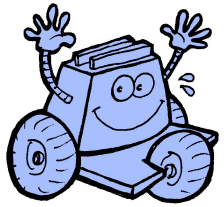
Motion Planning

Latombe (1991):

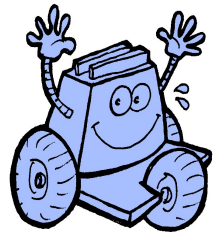
"…eminently necessary since, by definition, a robot accomplishes tasks by moving in the real world."

Goals:

- Collision-free trajectories.
- Robot should reach the goal location as fast as possible.

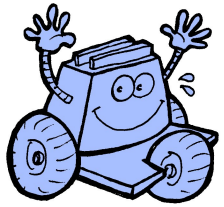


Motion Planning in Dynamic Environments

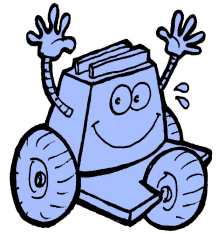


- How to react to unforeseen obstacles?
  - efficiency
  - reliability

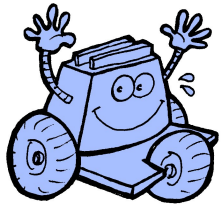
- Dynamic Window Approaches
  [Simmons, 96], [Fox et al., 97], [Brock & Khatib, 99]
- Grid map based planning
  [Konolige, 00]
- Nearness Diagram Navigation
  [Minguez at al., 2001, 2002]
- Vector-Field-Histogram+
  [Ulrich & Borenstein, 98]
- A*, D*, D* Lite, ARA*, …



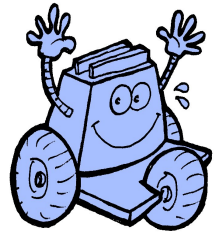
Two Challenges for Optimal Path planning



- Calculate the optimal path taking potential uncertainties in the actions into account
- Quickly generate actions in the case of unforeseen objects



Classic Two-Layered Architecture for Mobile Robots



map

sensor data

Planning

low frequency

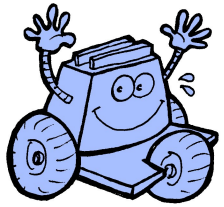
sub-goal

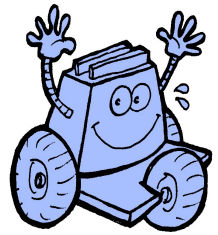
Collision Avoidance

high frequency

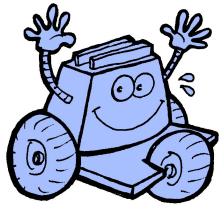
motion command

robot

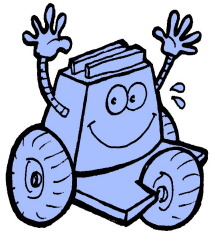




Dynamic Window Approach (DWA)



Dynamic Window Approach

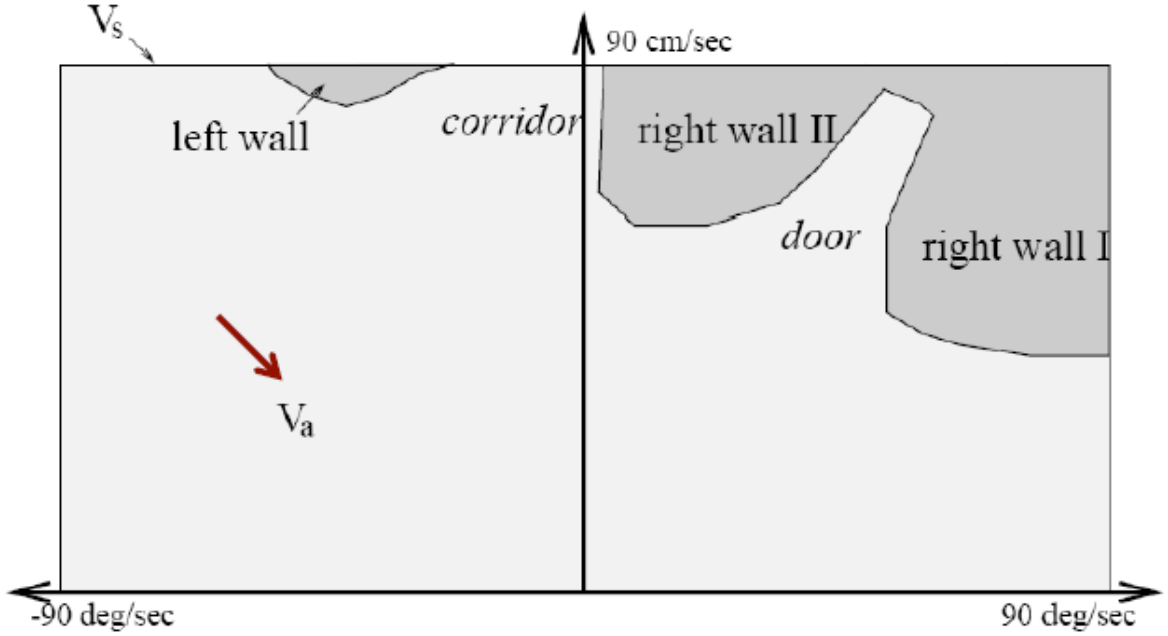


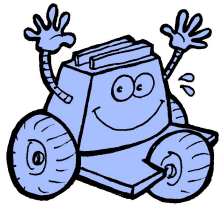
- **Collision avoidance:** Determine collision-free trajectories using geometric operations
- Here: Robot moves on circular arcs
- Motion commands (v,ω)
- Which (v,ω) are admissible and reachable?

Admissible Velocities

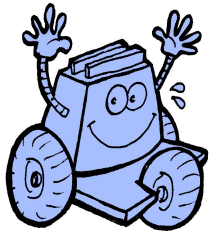
- Speeds are admissible if the robot would be able to stop before reaching the obstacle

$$V_a = \{(v,\omega) \mid v \leq \sqrt{2\text{dist}(v,\omega)a_{trans}} \wedge \omega \leq \sqrt{2\text{dist}(v,\omega)a_{rot}}\}$$



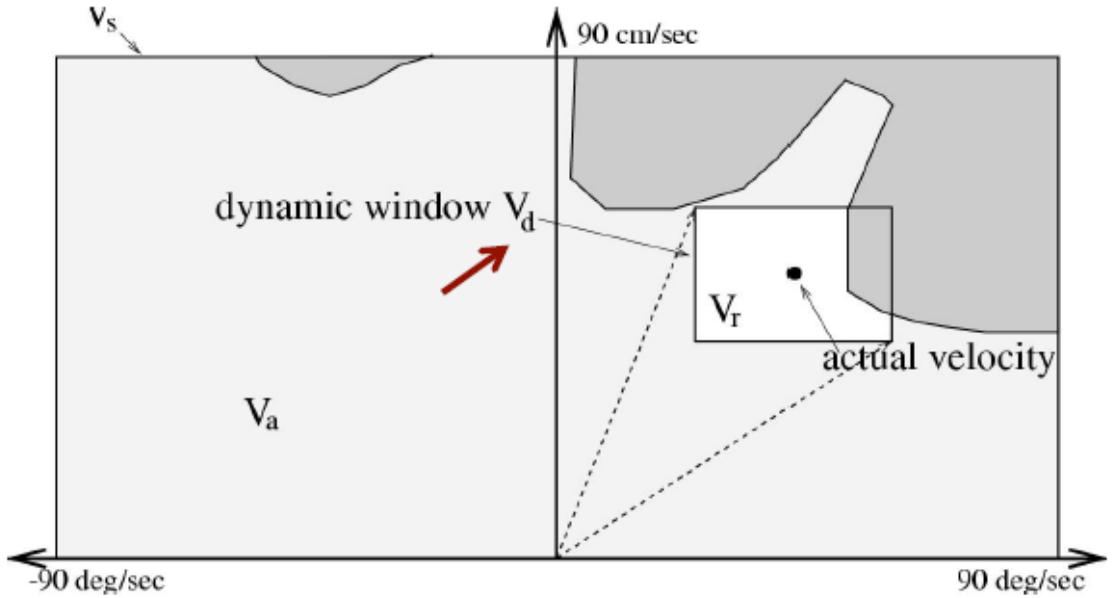


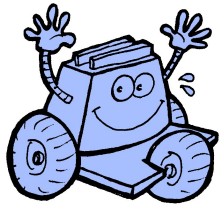
Reachable Velocities

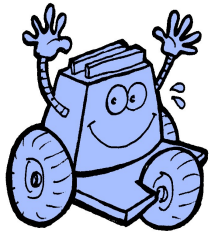


- Speeds that are reachable by acceleration

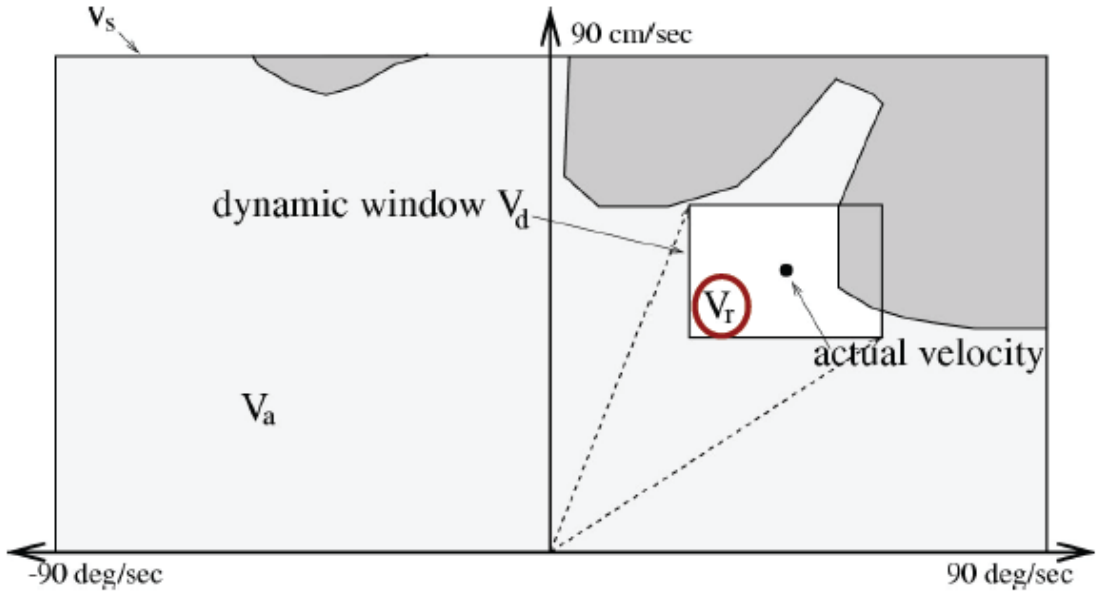
$$V_d = \{(v, \omega) \mid v \in [v - a_{trans}t, v + a_{trans}t] \wedge \omega \in [\omega - a_{rot}t, \omega + a_{rot}t]\}$$





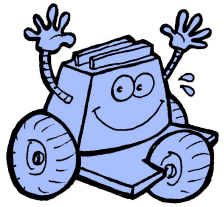


DWA Search Space

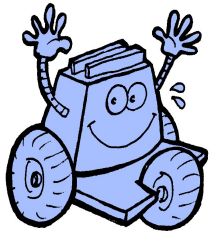


- V_s = all possible speeds of the robot.
- V_a = obstacle free area.
- V_d = speeds reachable within a certain time frame based on possible accelerations.

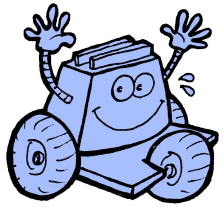
$$V_r = V_s \cap V_a \cap V_d$$



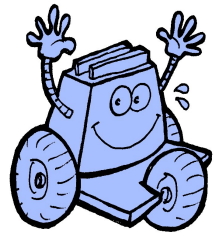
Dynamic Window Approach



- How to choose <v,ω>?
- Steering commands are chosen by a heuristic navigation function.
- This function tries to minimize the travel-time by:
  "**driving fast** in the **right direction.**"



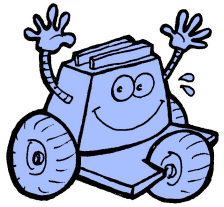
Dynamic Window Approach



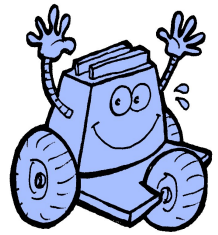
- Heuristic navigation function.
- Planning restricted to <x,y>-space.
- No planning in the velocity space.

Navigation Function: [Brock & Khatib, 99]

$$NF = \alpha \cdot vel + \beta \cdot nf + \gamma \cdot \Delta nf + \delta \cdot goal$$



Dynamic Window Approach

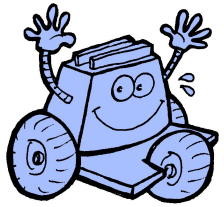


- Heuristic navigation function.
- Planning restricted to <x,y>-space.
- No planning in the velocity space.

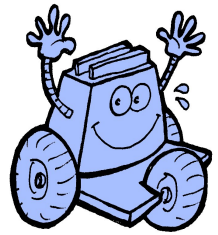
Navigation Function: [Brock & Khatib, 99]

$$NF = \alpha \cdot vel + \beta \cdot nf + \gamma \cdot \Delta nf + \delta \cdot goal$$

Maximizes velocity.



Dynamic Window Approach



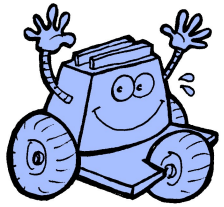
- Heuristic navigation function.
- Planning restricted to <x,y>-space.
- No planning in the velocity space.

Navigation Function: [Brock & Khatib, 99]

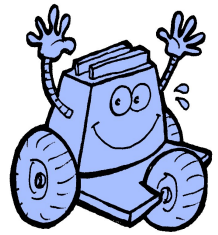
$$NF = \alpha \cdot vel + \beta \cdot nf + \gamma \cdot \Delta nf + \delta \cdot goal$$

Maximizes velocity.

Considers cost to reach the goal.



Dynamic Window Approach



- Heuristic navigation function.
- Planning restricted to <x,y>-space.
- No planning in the velocity space.

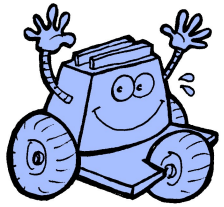
Navigation Function: [Brock & Khatib, 99]

$$NF = \alpha \cdot vel + \beta \cdot nf + \gamma \cdot \Delta nf + \delta \cdot goal$$

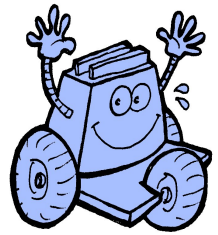
Maximizes velocity.

Considers cost to reach the goal.

Follows grid based path computed by A*.



Dynamic Window Approach



- Heuristic navigation function.
- Planning restricted to <x,y>-space.
- No planning in the velocity space.

Navigation Function:

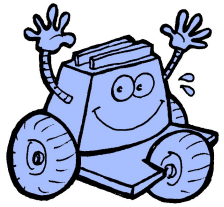
Goal nearness.

$$NF = \alpha \cdot vel + \beta \cdot nf + \gamma \cdot \Delta nf + \delta \cdot goal$$

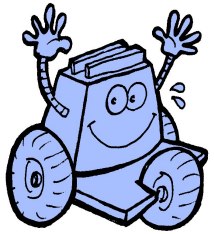
Maximizes velocity.

Considers cost to reach the goal.

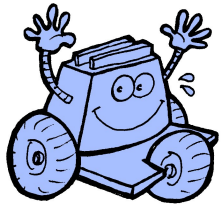
Follows grid based path computed by A*.

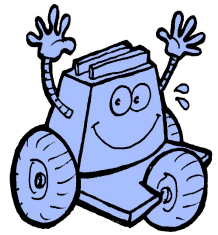


Dynamic Window Approach: PLUSES of this method

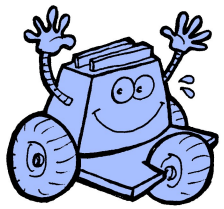


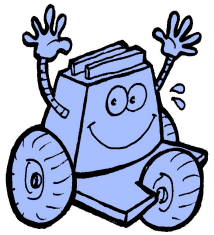
- Reacts quickly.
- Low CPU power requirements.
- Guides a robot on a collision-free path.
- Successfully used in a lot of real-world scenarios.
- Resulting trajectories sometimes sub-optimal.
- Local minima might prevent the robot from reaching the goal location.



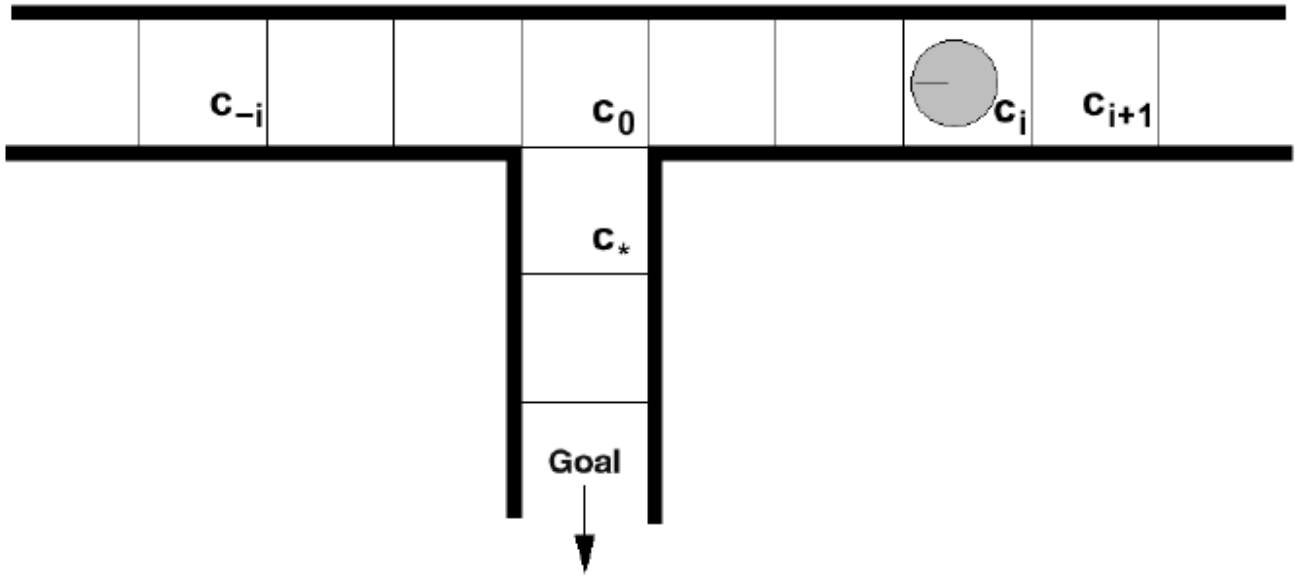


Problems of DWA

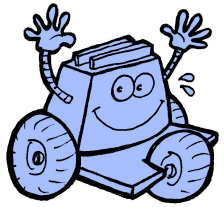


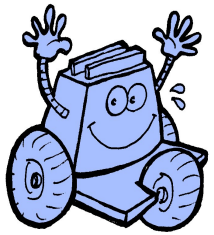


Problems of DWAs

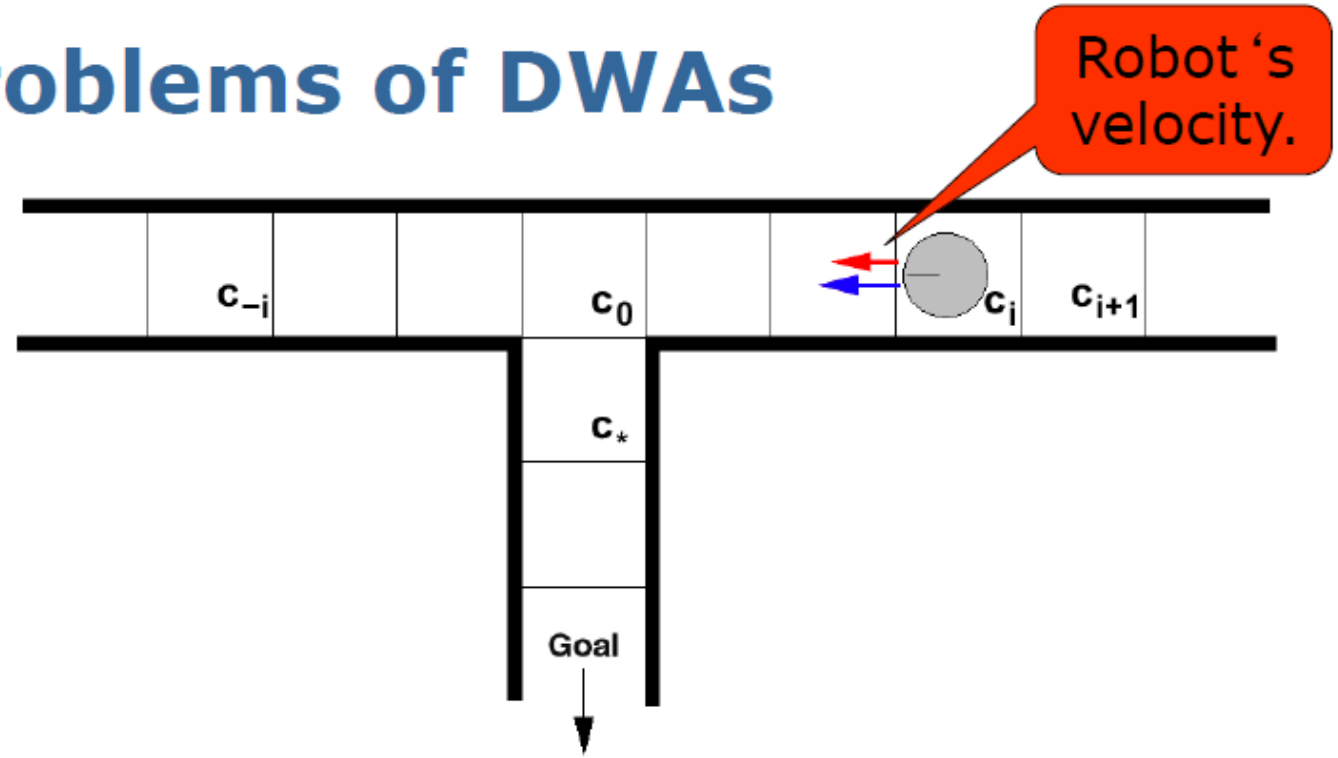


$$NF = \alpha \cdot vel + \beta \cdot nf + \gamma \cdot \Delta nf + \delta \cdot goal$$

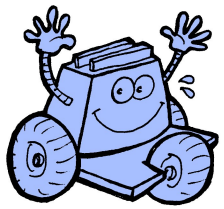


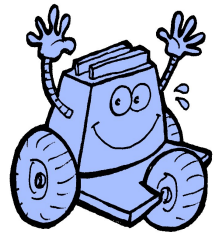


Problems of DWAs

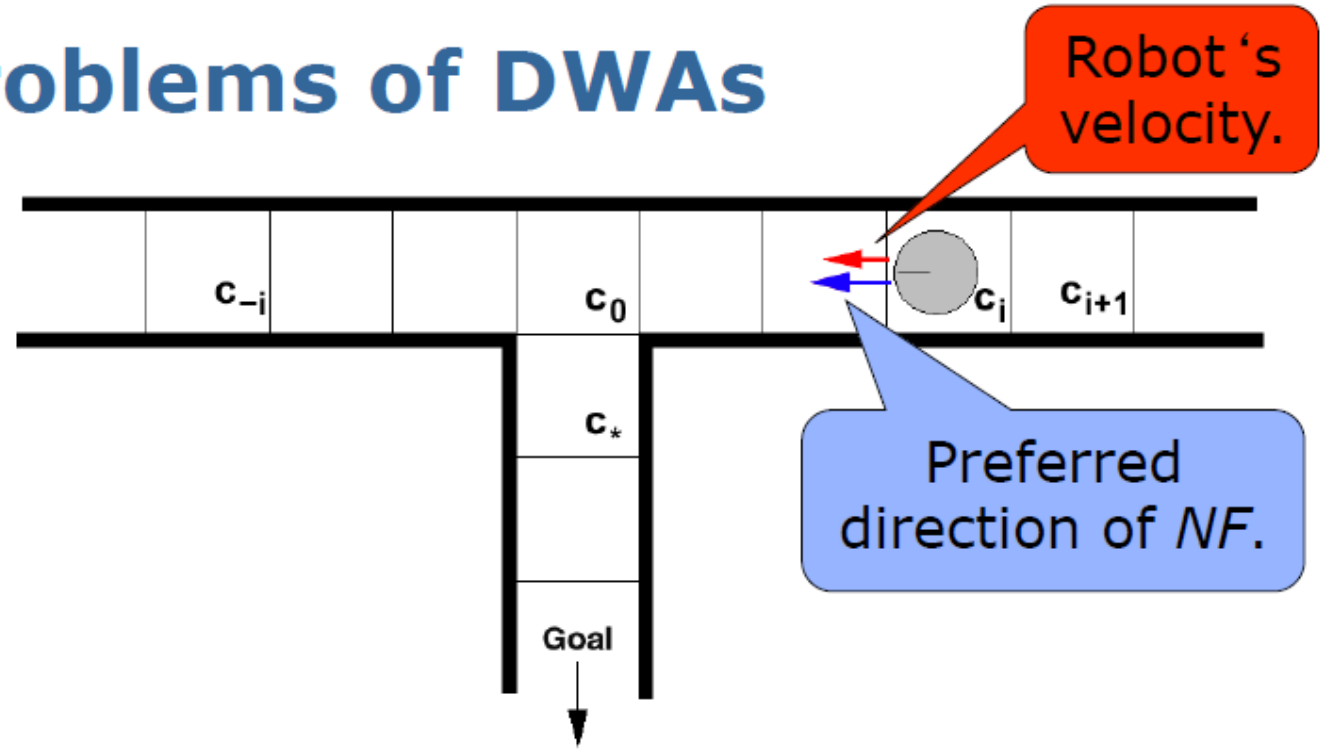


$$NF = \alpha \cdot vel + \beta \cdot nf + \gamma \cdot \Delta nf + \delta \cdot goal$$

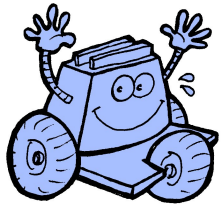


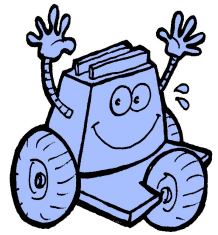


Problems of DWAs

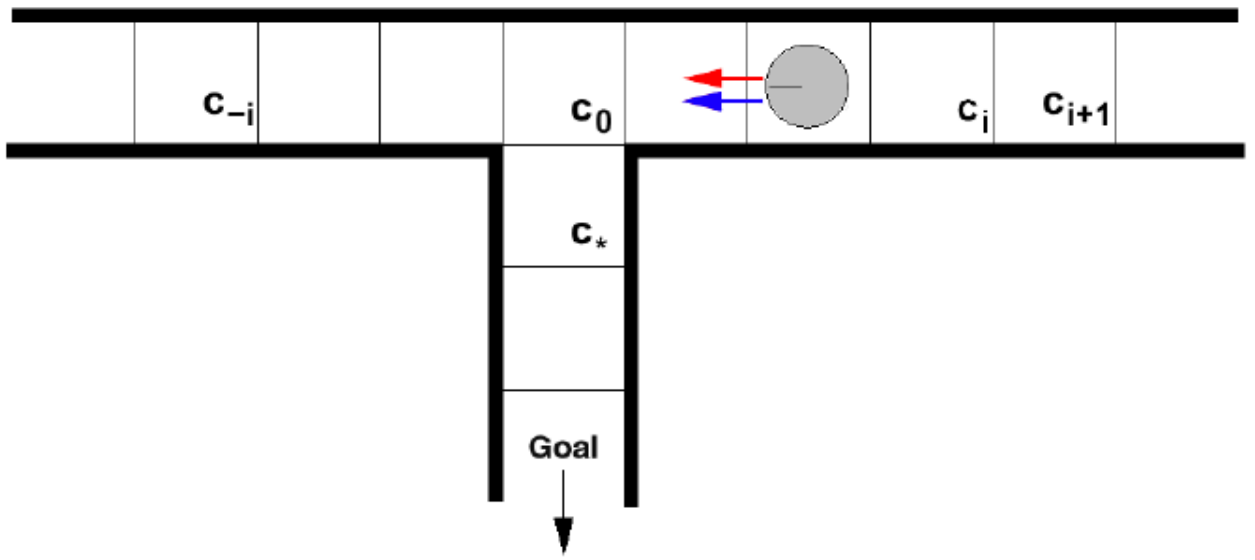


$$NF = \alpha \cdot vel + \beta \cdot nf + \gamma \cdot \Delta nf + \delta \cdot goal$$

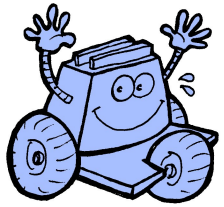


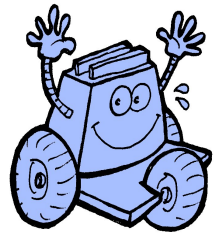


Problems of DWAs

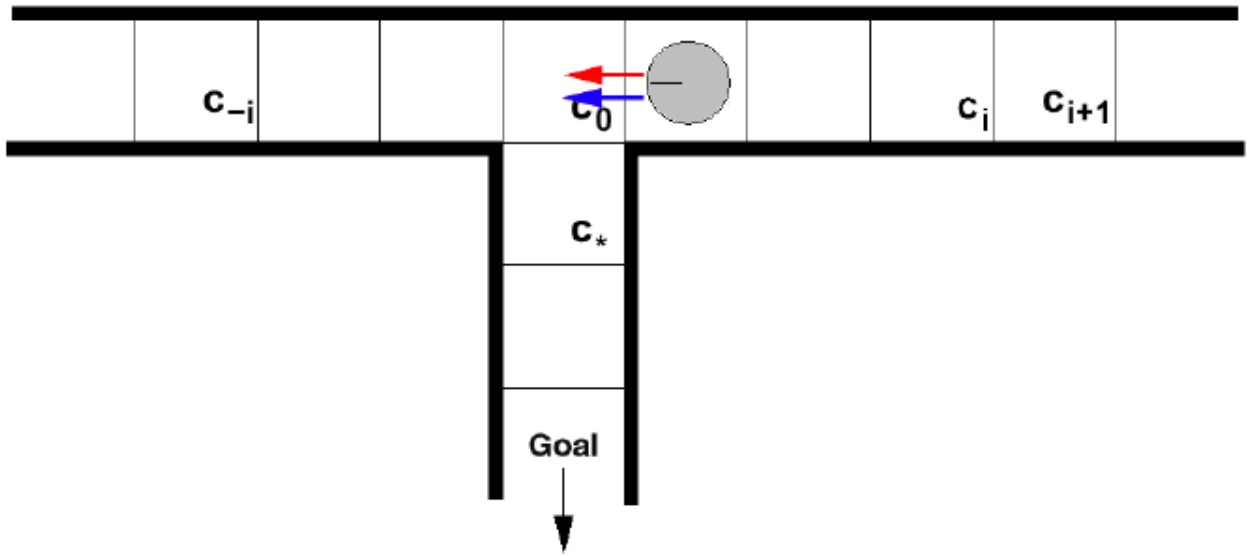


$$NF = \alpha \cdot vel + \beta \cdot nf + \gamma \cdot \Delta nf + \delta \cdot goal$$

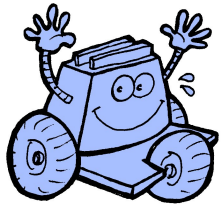


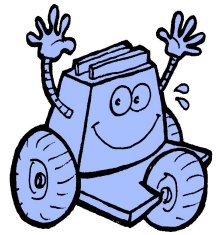


Problems of DWAs

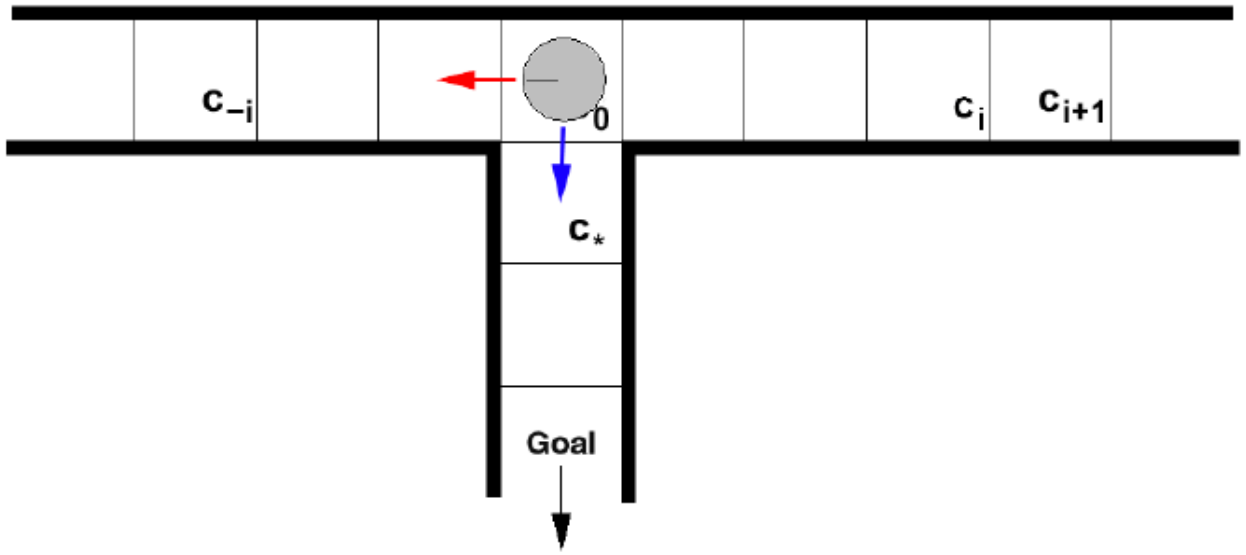


$$NF = \alpha \cdot vel + \beta \cdot nf + \gamma \cdot \Delta nf + \delta \cdot goal$$



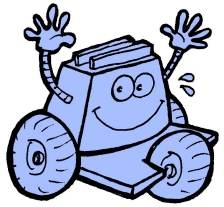


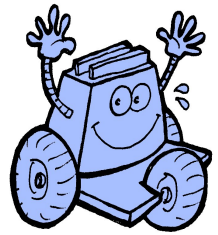
Problems of DWAs



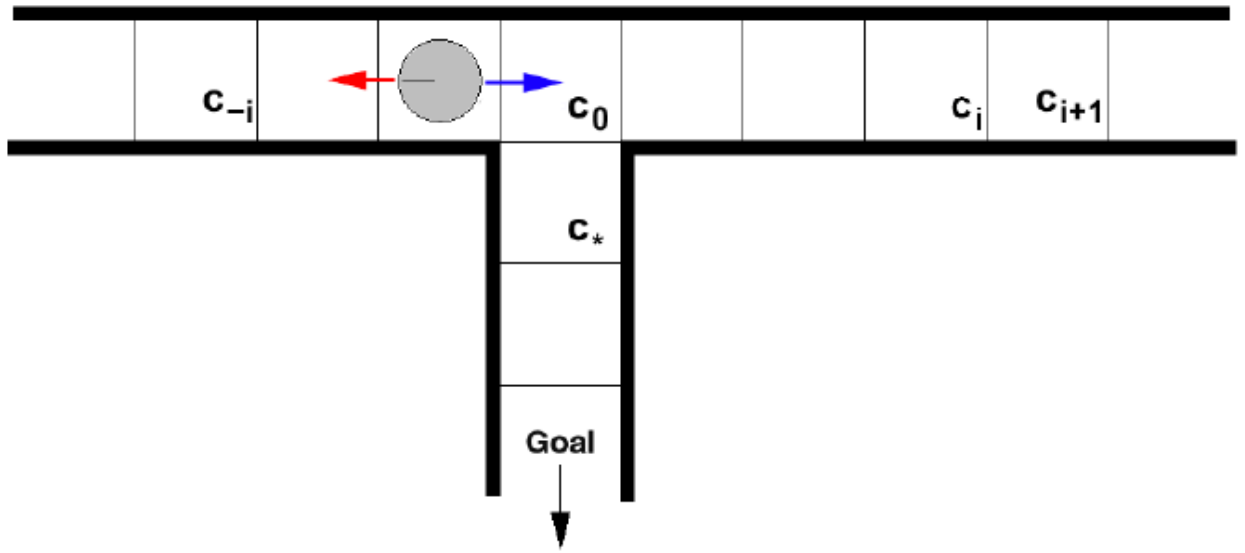
$$NF = \alpha \cdot vel + \beta \cdot nf + \gamma \cdot \Delta nf + \delta \cdot goal$$

- The robot drives too fast at c_0 to enter corridor facing south.

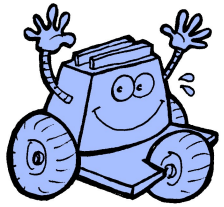


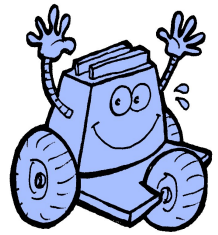


Problems of DWAs

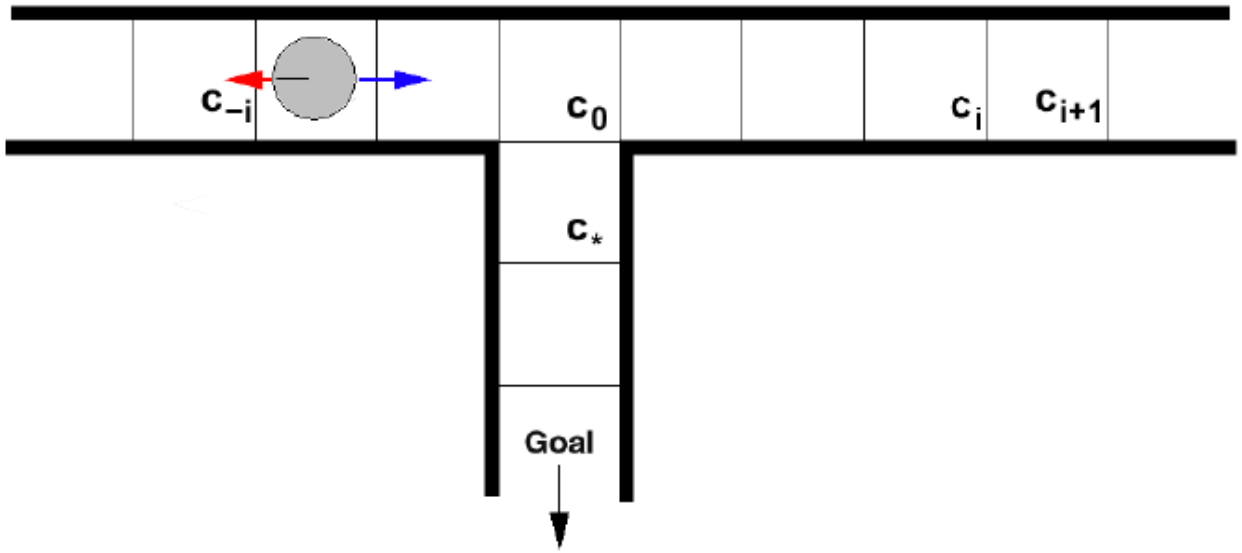


$$NF = \alpha \cdot vel + \beta \cdot nf + \gamma \cdot \Delta nf + \delta \cdot goal$$

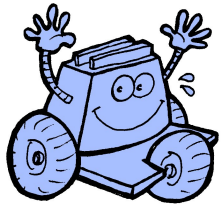


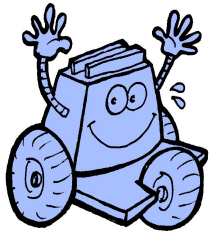


Problems of DWAs

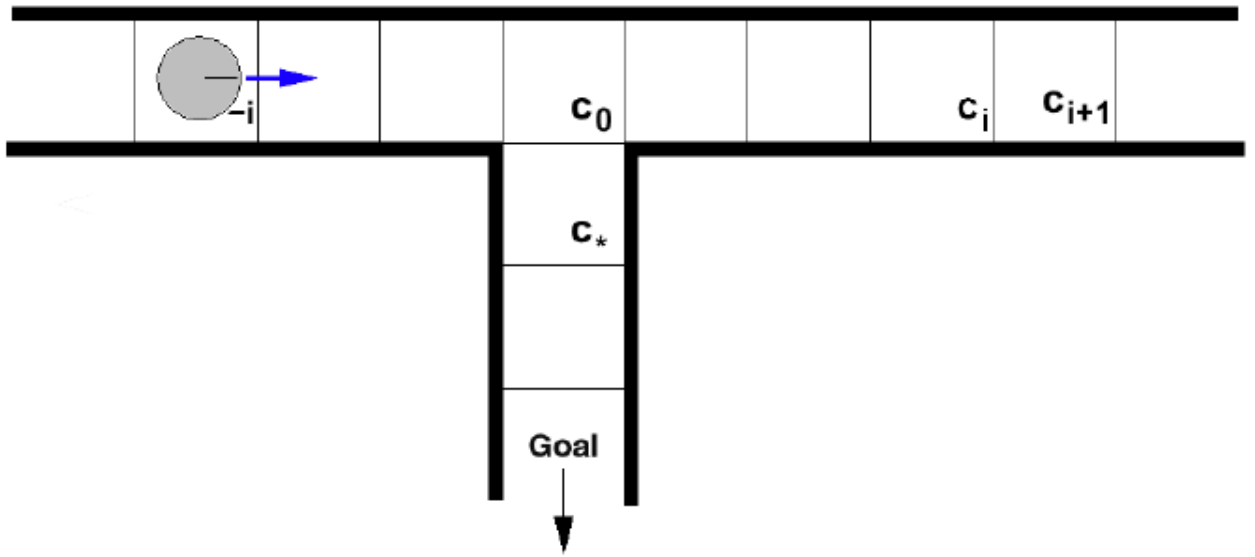


$$NF = \alpha \cdot vel + \beta \cdot nf + \gamma \cdot \Delta nf + \delta \cdot goal$$



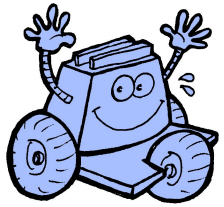


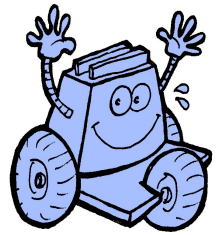
Problems of DWAs



- Same situation as in the beginning.

  ➔ DWAs have problems to reach the goal.

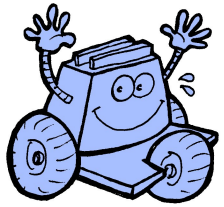




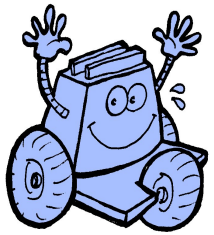
Problems of DWAs

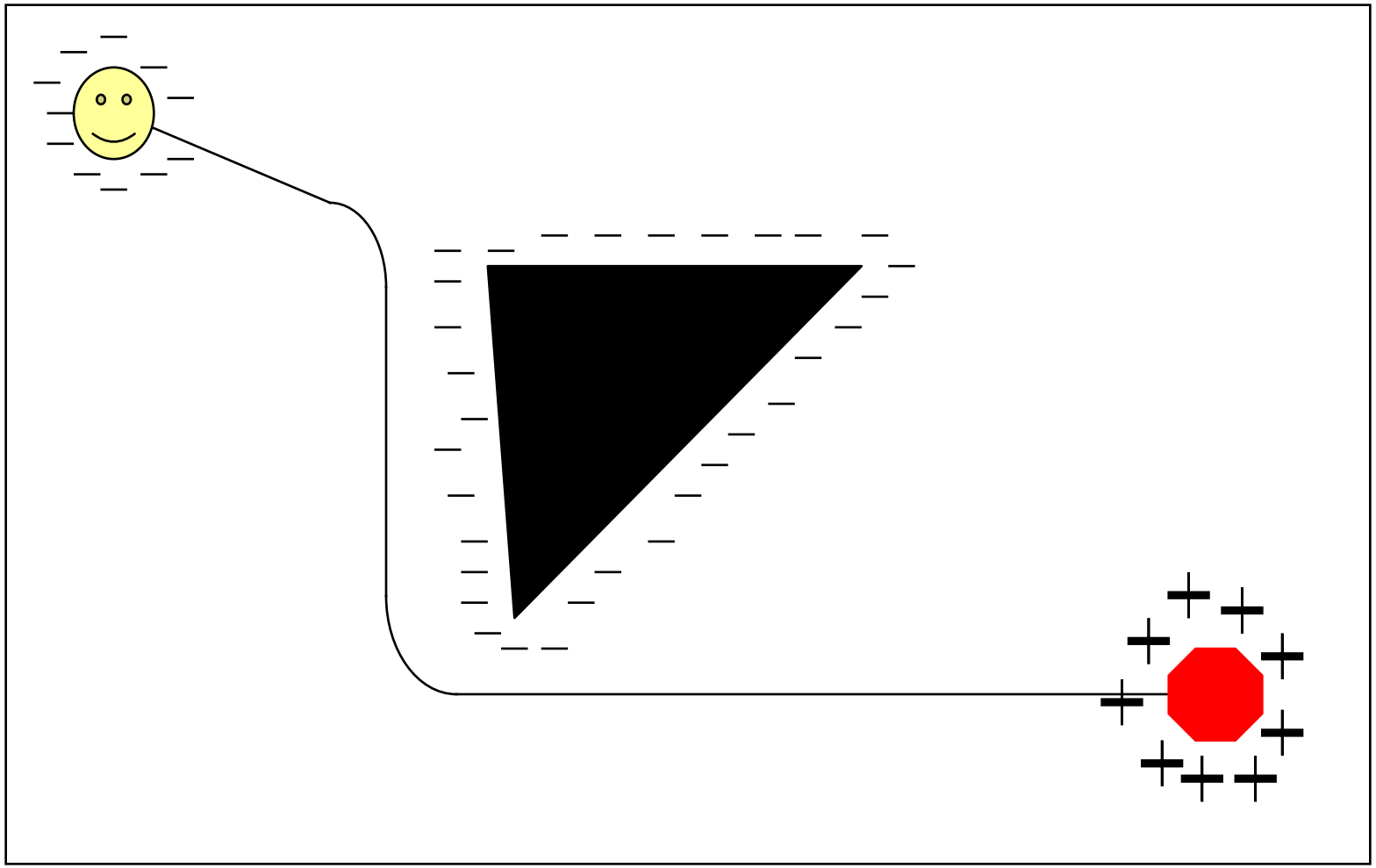
- Typical problem in a real world situation:

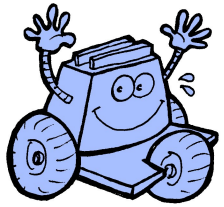
- Robot does not slow down early enough to enter the doorway.



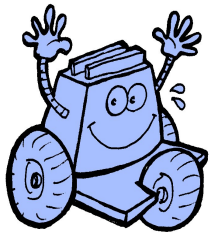
Potential Field Method





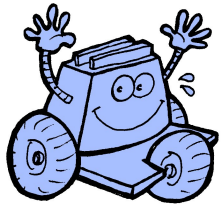


Potential Field Methods

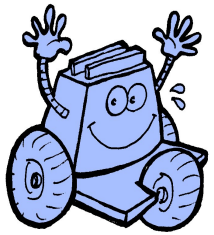


Basic Idea:

- robot is represented by a point in C-space
- treat robot like particle under the influence of an **artificial potential field U**
- **U** is constructed to reflect (locally) the structure of the free C-space (hence called 'local' methods)
- originally proposed by Khatib for on-line collision avoidance for a robot with proximity sensors

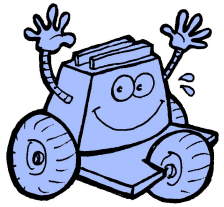


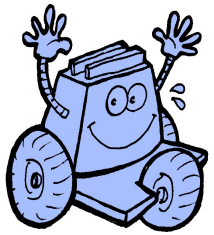
Potential Field



The Potential Function: $\mathbf{U} : \mathcal{C}_{free} \longrightarrow \mathbb{R}^1$

- want robot to be *attacted* to goal and *repelled* from obstacles
  - **attractive potential** $\mathbf{U}_{att}(\mathbf{q})$ associated with $\mathbf{q}_{goal}$
  - **repulsive potential** $\mathbf{U}_{rep}(\mathbf{q})$ associated with $\mathcal{CB}$
  - $\mathbf{U}(\mathbf{q}) = \mathbf{U}_{att}(\mathbf{q}) + \mathbf{U}_{rep}(\mathbf{q})$
- $\mathbf{U}(\mathbf{q})$ must be differentiable for every $\mathbf{q} \in \mathcal{C}_{free}$





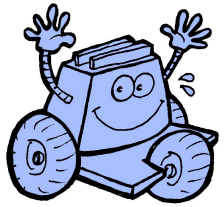
The Field of Artificial Forces

The Field of Artificial Forces: $\vec{F}(\mathbf{q}) = -\nabla \mathbf{U}(\mathbf{q})$

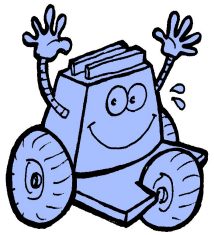
- $\nabla \mathbf{U}(\mathbf{q})$ denotes gradient of $\mathbf{U}$ at $\mathbf{q}$, i.e., $\nabla \mathbf{U}(\mathbf{q})$ is a vector that 'points' in the direction of 'fastest change' of $\mathbf{U}$ at configuration $\mathbf{q}$
- e.g., if $\mathcal{W} = \mathbb{R}^2$, then $\mathbf{q} = (x, y)$ and

$$\nabla \mathbf{U}(\mathbf{q}) = \begin{bmatrix} \frac{\partial \mathrm{U}}{\partial x} \\ \\ \frac{\partial \mathrm{U}}{\partial y} \end{bmatrix}$$

- $|\nabla \mathbf{U}(\mathbf{q})| = \sqrt{(\frac{\partial \mathrm{U}}{\partial x})^2 + (\frac{\partial \mathrm{U}}{\partial y})^2}$ is the magnitude of the rate of change
- $\vec{F}(\mathbf{q}) = -\nabla \mathbf{U}_{att}(\mathbf{q}) - \nabla \mathbf{U}_{rep}(\mathbf{q})$

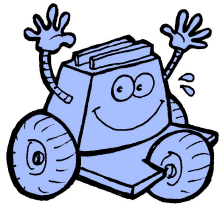


The Attractive Potential

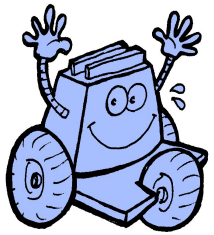


The Attractive Potential

Basic Idea: $\mathbf{U}_{att}(\mathbf{q})$ should **increase** as $\mathbf{q}$ moves **away from** $\mathbf{q}_{goal}$ (like potential energy increases as you move away from earth's surface)



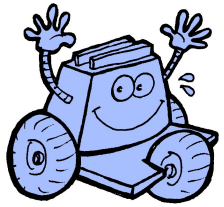
The Attractive Potential



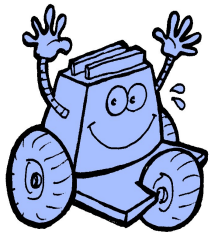
Better Idea: $\mathbf{U}_{att}(\mathbf{q})$ is a 'parabolic well'

- $\mathbf{U}_{att}(\mathbf{q}) = \frac{1}{2}\xi\rho^2_{goal}(\mathbf{q})$, where
  - $\rho_{goal}(\mathbf{q}) = \|\mathbf{q} - \mathbf{q}_{goal}\|$, i.e., Euclidean distance
  - ξ is some positive constant scaling factor
- unique minimum at $\mathbf{q}_{goal}$, i.e., $\mathbf{U}_{att}(\mathbf{q}_{goal}) = 0$
- $\mathbf{U}_{att}(\mathbf{q})$ differentiable for all $\mathbf{q}$

$$\begin{aligned}\vec{F}_{att}(\mathbf{q}) = -\nabla\mathbf{U}_{att}(\mathbf{q}) &= -\nabla\frac{1}{2}\xi\rho^2_{goal}(\mathbf{q}) \\ &= -\frac{1}{2}\xi\nabla\rho^2_{goal}(\mathbf{q}) \\ &= -\frac{1}{2}\xi(2\rho_{goal}(\mathbf{q}))\nabla\rho_{goal}(\mathbf{q})\end{aligned}$$



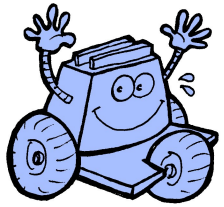
The Attractive Potential



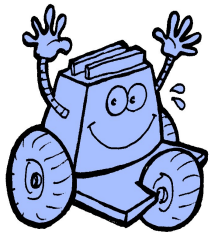
Recall: $\rho_{goal}(\mathbf{q}) = \|\mathbf{q} - \mathbf{q}_{goal}\| = (\Sigma_i (x_i - x_{g_i})^2)^{1/2}$,
where $\mathbf{q} = (x_1, \ldots, x_n)$ and $\mathbf{q}_{goal} = (x_{g_1}, \ldots, x_{g_n})$

$$
\begin{aligned}
\nabla \rho_{goal}(\mathbf{q}) &= \nabla \left(\sum_i (x_i - x_{g_i})^2 \right)^{1/2} \\
&= \frac{1}{2} \left(\sum_i (x_i - x_{g_i})^2 \right)^{-1/2} \nabla \left(\sum_i (x_i - x_{g_i})^2 \right) \\
&= \frac{1}{2} \left(\sum_i (x_i - x_{g_i})^2 \right)^{-1/2} (2(x_1 - x_{g_1}), \ldots, 2(x_n - x_{g_n})) \\
&= \frac{(x_1, \ldots, x_n) - (x_{g_1}, \ldots, x_{g_n})}{(\Sigma_i (x_i - x_{g_i})^2)^{1/2}} \\
&= \frac{\mathbf{q} - \mathbf{q}_{goal}}{\|\mathbf{q} - \mathbf{q}_{goal}\|} = \frac{\mathbf{q} - \mathbf{q}_{goal}}{\rho_{goal}(\mathbf{q})}
\end{aligned}
$$

So, $-\nabla \rho_{goal}(\mathbf{q})$ is a unit vector directed toward $\mathbf{q}_{goal}$ from $\mathbf{q}$



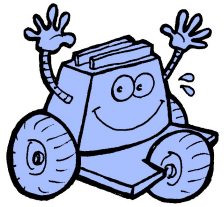
The Attractive Potential



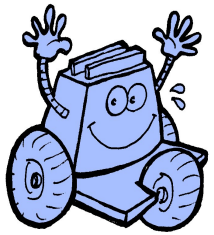
Thus, since $-\nabla \mathbf{U}_{att}(\mathbf{q}) = -\frac{1}{2}\xi(2\rho_{goal}(\mathbf{q}))\nabla\rho_{goal}(\mathbf{q})$, we get:

$$\vec{F}_{att}(\mathbf{q}) = -\nabla \mathbf{U}_{att}(\mathbf{q}) = -\xi(\mathbf{q} - \mathbf{q}_{goal})$$

- $\vec{F}_{att}(\mathbf{q})$ is a vector directed toward $\mathbf{q}_{goal}$ with magnitude linearly related to the distance from $\mathbf{q}$ to $\mathbf{q}_{goal}$
- $\vec{F}_{att}(\mathbf{q})$ converges linearly to zero as $\mathbf{q}$ approaches $\mathbf{q}_{goal}$ – good for stability
- $\vec{F}_{att}(\mathbf{q})$ grows without bound as $\mathbf{q}$ moves away from $\mathbf{q}_{goal}$ – not so good

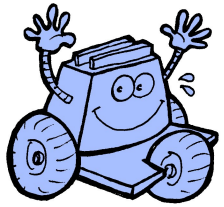


The Repulsive Potential

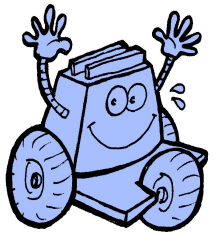


Basic Idea: $\mathcal{A}$ should be repelled from obstacles

- never want to let $\mathcal{A}$ 'hit' an obstacle
- if $\mathcal{A}$ is far from obstacle, don't want obstacle to affect $\mathcal{A}$'s motion



The Repulsive Potential



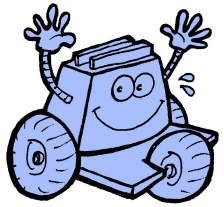
One Choice for $\mathbf{U}_{rep}$:

$$\mathbf{U}_{rep}(\mathbf{q}) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho(\mathbf{q})} - \frac{1}{\rho_0}\right) & \text{if } \rho(\mathbf{q}) \leq \rho_0 \\ 0 & \text{if } \rho(\mathbf{q}) > \rho_0 \end{cases}$$

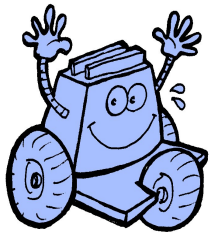
where

- $\rho(\mathbf{q})$ is minimum distance from $\mathcal{CB}$ to $\mathbf{q}$, i.e., $\rho(\mathbf{q}) = \min_{\mathbf{q}' \in \mathcal{CB}} \|\mathbf{q} - \mathbf{q}'\|$
- η is a positive scaling factor
- ρ_0 is a positive constant – *distance of influence*

So, as $\mathbf{q}$ approaches $\mathcal{CB}$, $\mathbf{U}_{rep}(\mathbf{q})$ approaches ∞

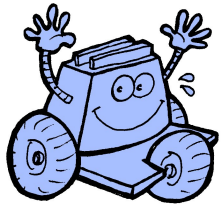


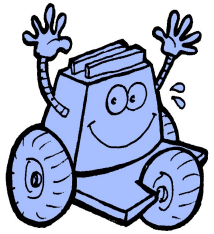
The Repulsive Potential



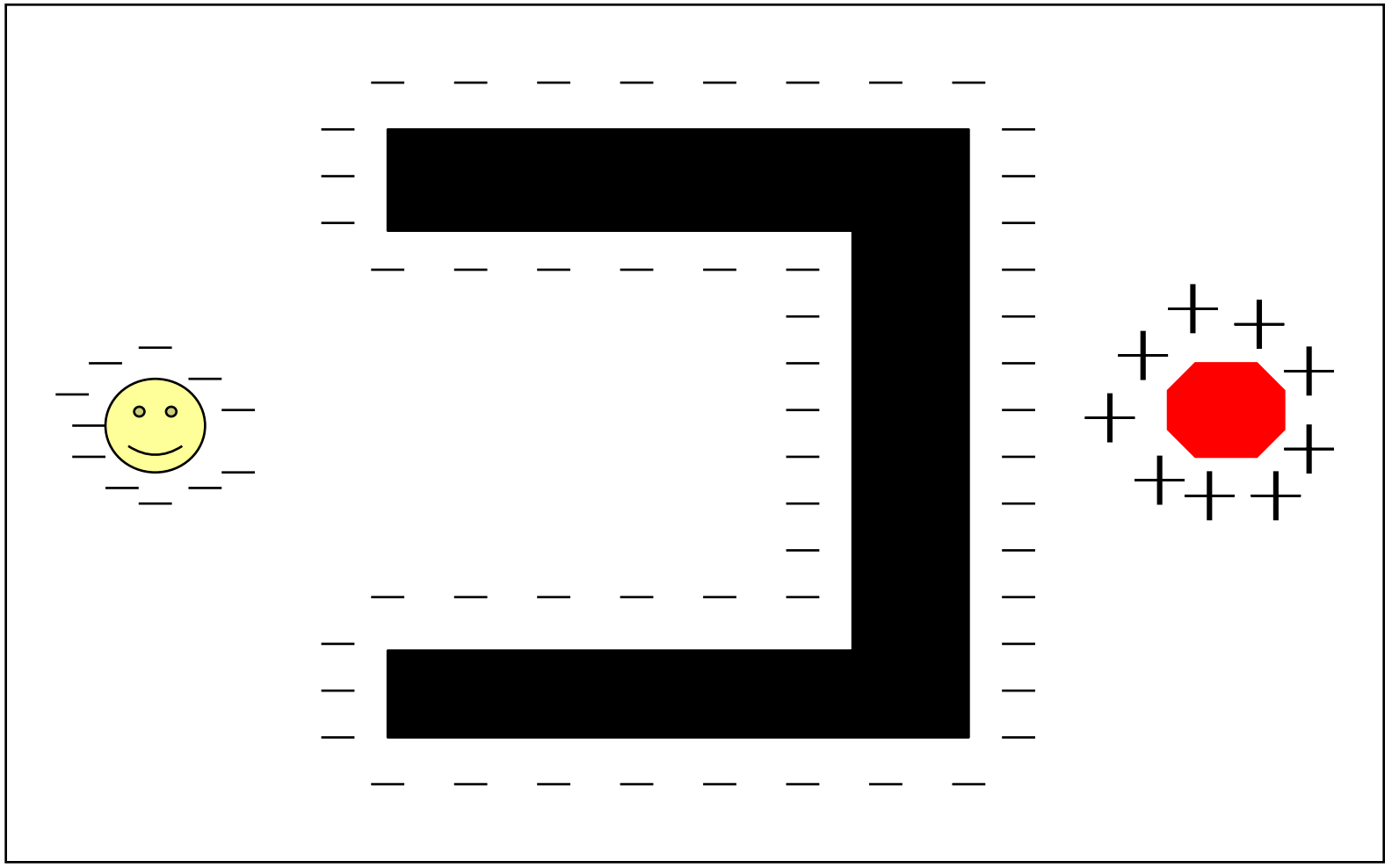
on designing $\mathbf{U}_{rep}$

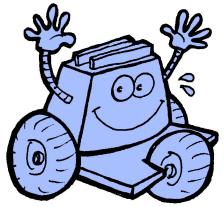
- can select different η and ρ_0 for each obstacle region – ρ_0 small for $\mathcal{CB}_i$ close to goal (or else repulsive force may keep us from ever reaching goal)
- if $\mathbf{U}_{rep}(\mathbf{q}_{goal}) \neq 0$, then global minimum of $\mathbf{U}(\mathbf{q})$ is generally not at $\mathbf{q}_{goal}$



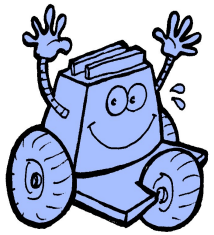


Local Minimum Problem with the Charge Analogy



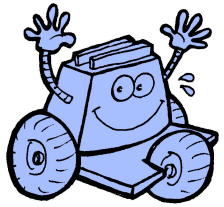


Escaping Local Minima - Random Motions

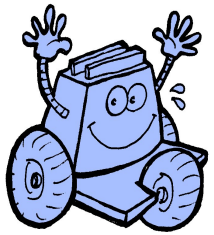


Overview [Barraquand and Latombe, 1989]

1. start at $\mathbf{q}_{init}$ and use best-first search to follow the gradient of $\mathbf{U}(\mathbf{q})$ until reach a local minimum $\mathbf{q}_{loc}$ (called **gradient motion**)
2. execute a **random walk** starting at $\mathbf{q}_{loc}$ and ending at some new configuration $\mathbf{q}'$ (random walk approximates Brownian motion)
3. execute another gradient motion starting $\mathbf{q}'$ (end of random walk)
4. repeat until reach $\mathbf{q}_{goal}$



Gradient Descent Planning



GRADIENT DESCENT PLANNING

input: $\mathbf{q}_{init}$, $\mathbf{q}_{goal}$, $\mathbf{U}(\mathbf{q}) = \mathbf{U}_{att}(\mathbf{q}) + \mathbf{U}_{rep}(\mathbf{q})$, and $\vec{F}(\mathbf{q}) = -\nabla \mathbf{U}(\mathbf{q})$

output: a path connecting $\mathbf{q}_{init}$ and $\mathbf{q}_{goal}$

1. let $\mathbf{q}_0 = \mathbf{q}_{init}$, $i = 0$
2. if $\mathbf{q}_i \neq \mathbf{q}_{goal}$

   then $\mathbf{q}_{i+1} = \mathbf{q}_i + \delta_i \frac{\vec{F}(\mathbf{q})}{\|\vec{F}(\mathbf{q})\|}$ {take a step of size δ_i in direction $\vec{F}(\mathbf{q})$}

   else stop
3. set $i = i + 1$ and goto step 2