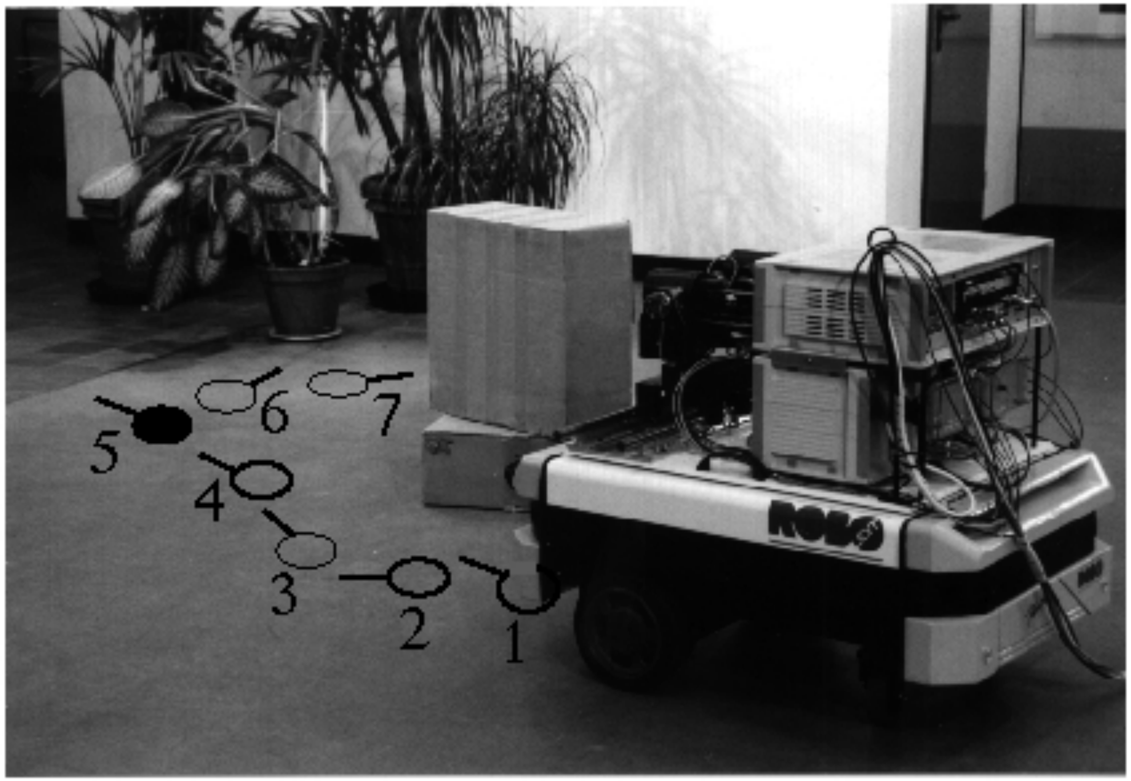
Vehicle motion planning and control: Survey of approaches

Gregor Schöner

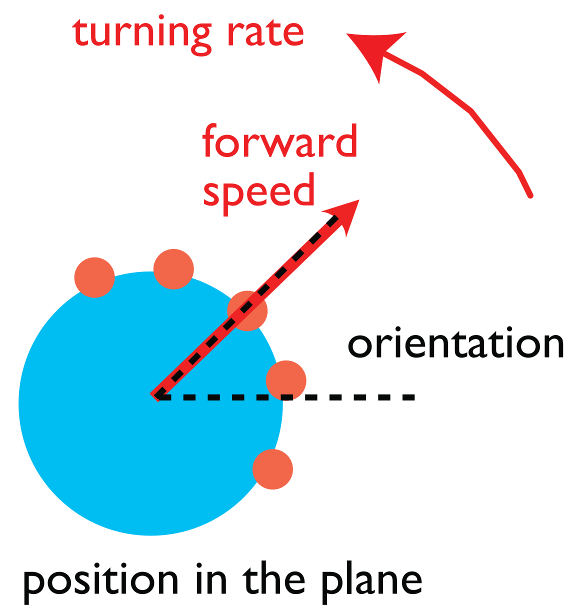
The problem

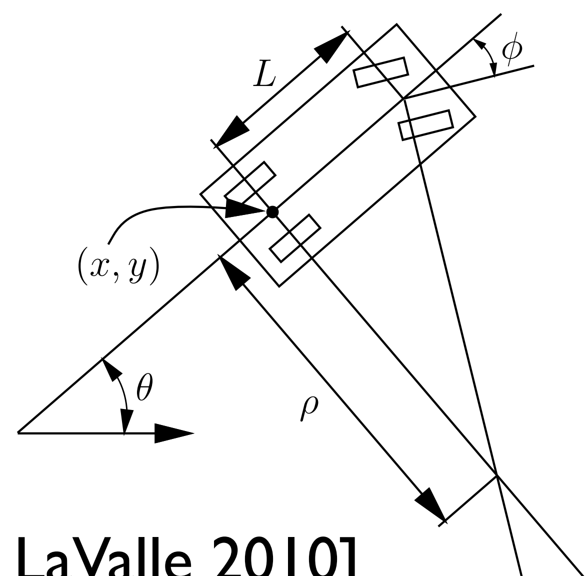
- move about in a 2D world, which is occupied by objects/stuff
- constraints
  - reach targets
  - avoid collisions
  - via points
  - orientations



Non-holonomic constraints

- Vehicles have typically non-holonomic constraints
  - fewer variables can be varied freely (e.g. velocities chosen) than variables that describe the physical state
  - state depends on the history of movement





[from LaValle 2010]

What is needed to autonomously move in an environment?

- sense something about the environment
- know about the environment
- plan movement in the environment that is collision-free
- control vehicle to achieve planned movement
- estimate what vehicle actually did

Concepts for planning

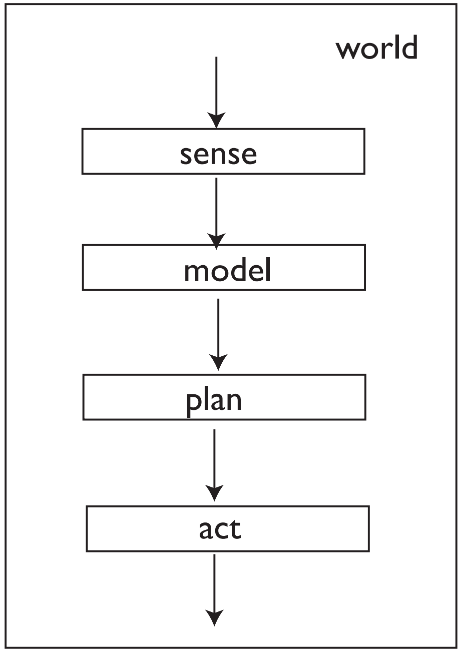
- local vs. global
  - planning based on information only about the local environment of the robot
  - vs. based on global map information about the environment
- reactive vs. planning
  - motion planning "on the fly" in response to sensory inputs
  - vs. motion planning for an entire action from initial to goal state

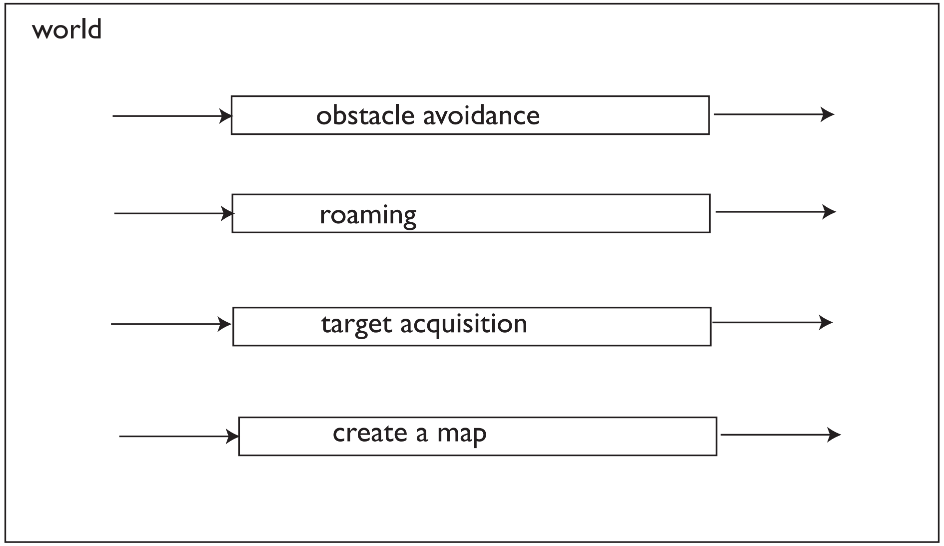
Concepts for planning

- exact vs. heuristic
  - exact: guarantee that a path that fulfills the constraints is found when one exists
  - vs. generate a plan based on ad hoc approach that is likely to fulfill constraints
- continuous vs. discrete:
  - continuous state space variables
  - vs. grid state spaces, graph state spaces

Concepts for planning

- sense-plan-act vs behavior-based
  - based on world representation that informs all planning
  - vs. based on low-level sensory information that is specific to each individual behavior, planning emerges from how behaviors interact



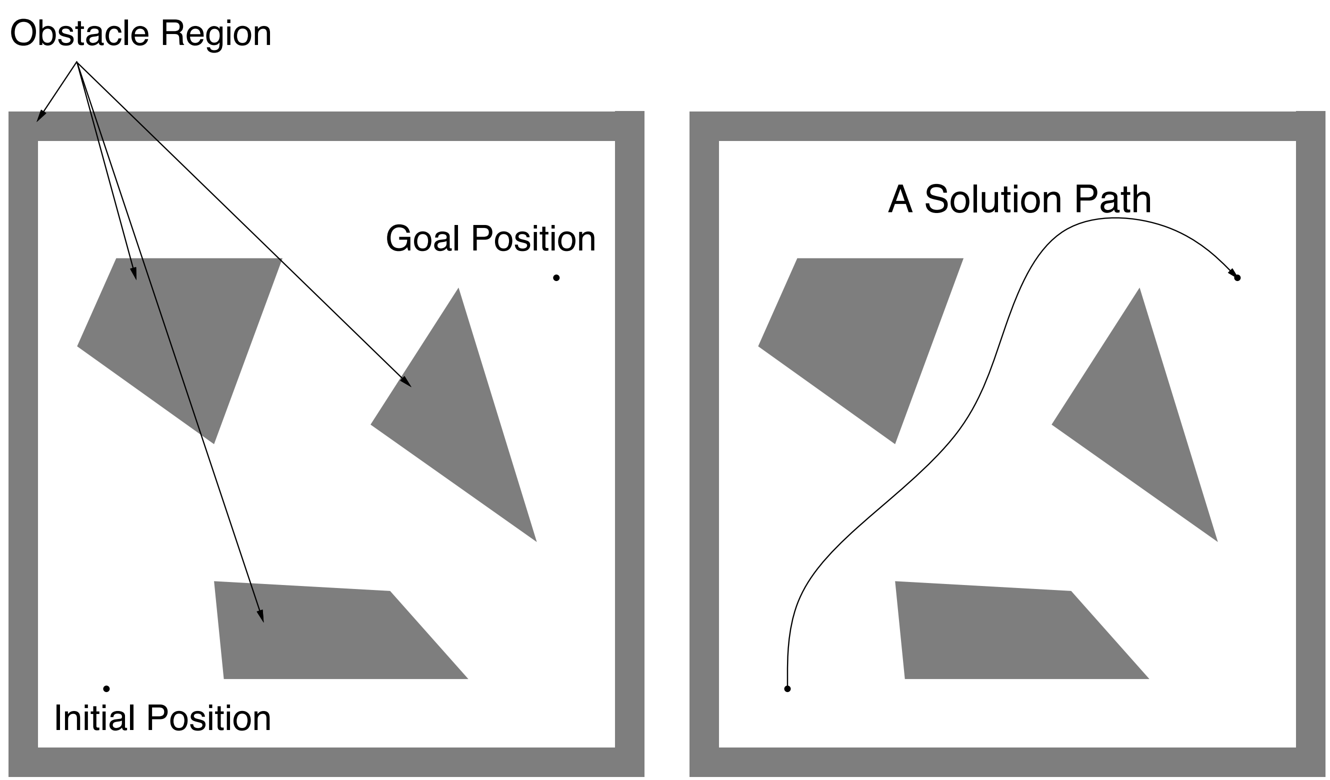


Approaches to vehicle path planning

- classical planning approaches
- potential field approach
- Borenstein & Koren
- Dynamic window approach
- (deliberate planning)

Classical global path planning

- standard reference: Latombe: Robot motion planning, 1991
- very good general review: LaValle: Planning algorithms, 2006, 2010



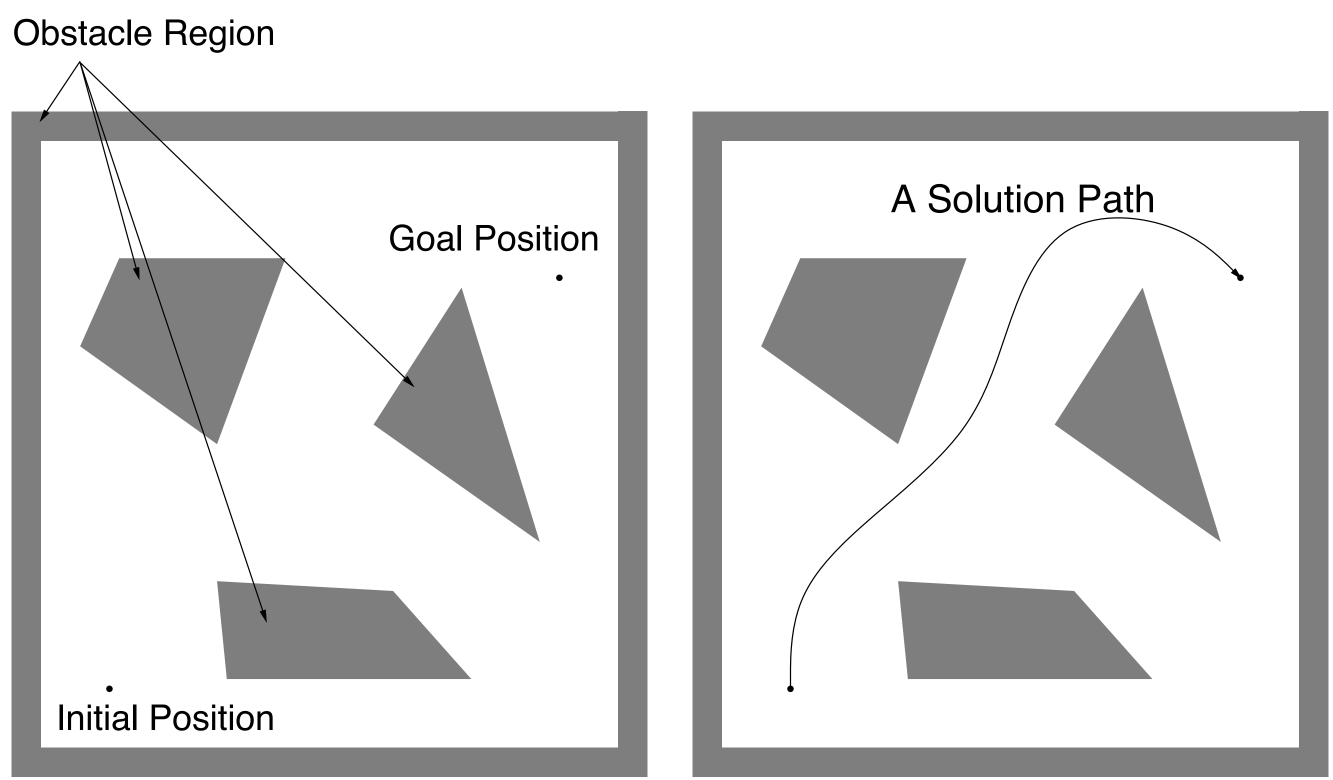
(a) (b)

[LaValle, 2006]

Classical global path planning

- mathematical theories of constraint satisfaction and decision theory
- searching spaces, sampling approaches

[LaValle, 2006]



(a) (b)

Classical local path planning

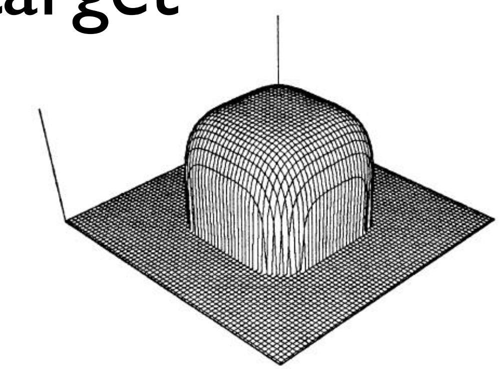
- reference: Cox, Wilfong: Autonomous Robot Vehicles, 1990
- based on a known world (e.g., represented as a polygonial model of surfaces)
- taking into account a kinematic model of the vehicle
- add smoothness constraints

Potential field approach

- invented by Khatib, 1986 (similar earlier formulation: Neville Hogan's impedance control)
- the trajectory of a manipulator or robot vehicle is generated by moving in a potential field to a minimum
- the manipulator 3D end-position or vehicle 2D position is updated by descending within that potential field
- obstacles are modeled as hills of potential field; target states are valleys/minima of the potential field

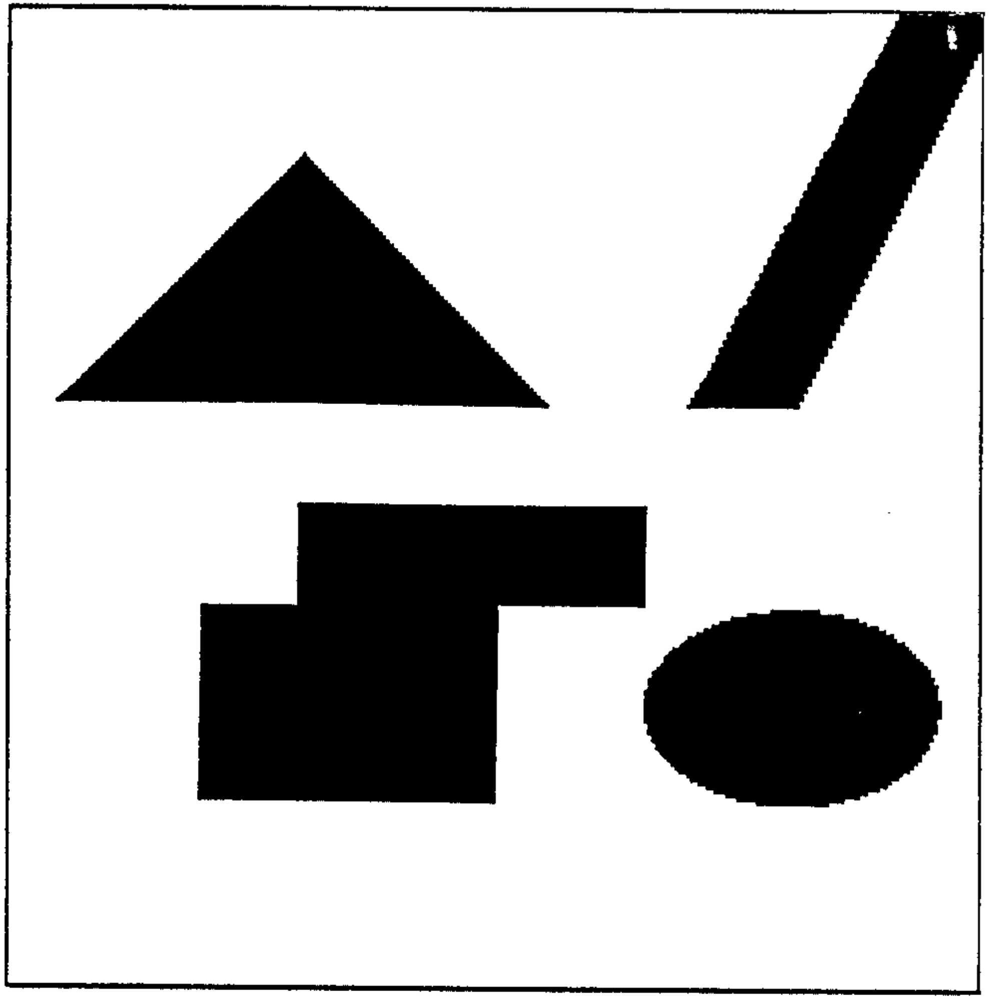
Potential field approach as a heuristic planning approach

- need a mathematical representation of target and obstacle configuration
- make potential minimum at target
- make potential maximum at obstacles
- compute downhill gradient descent for path generation



Potential field approach

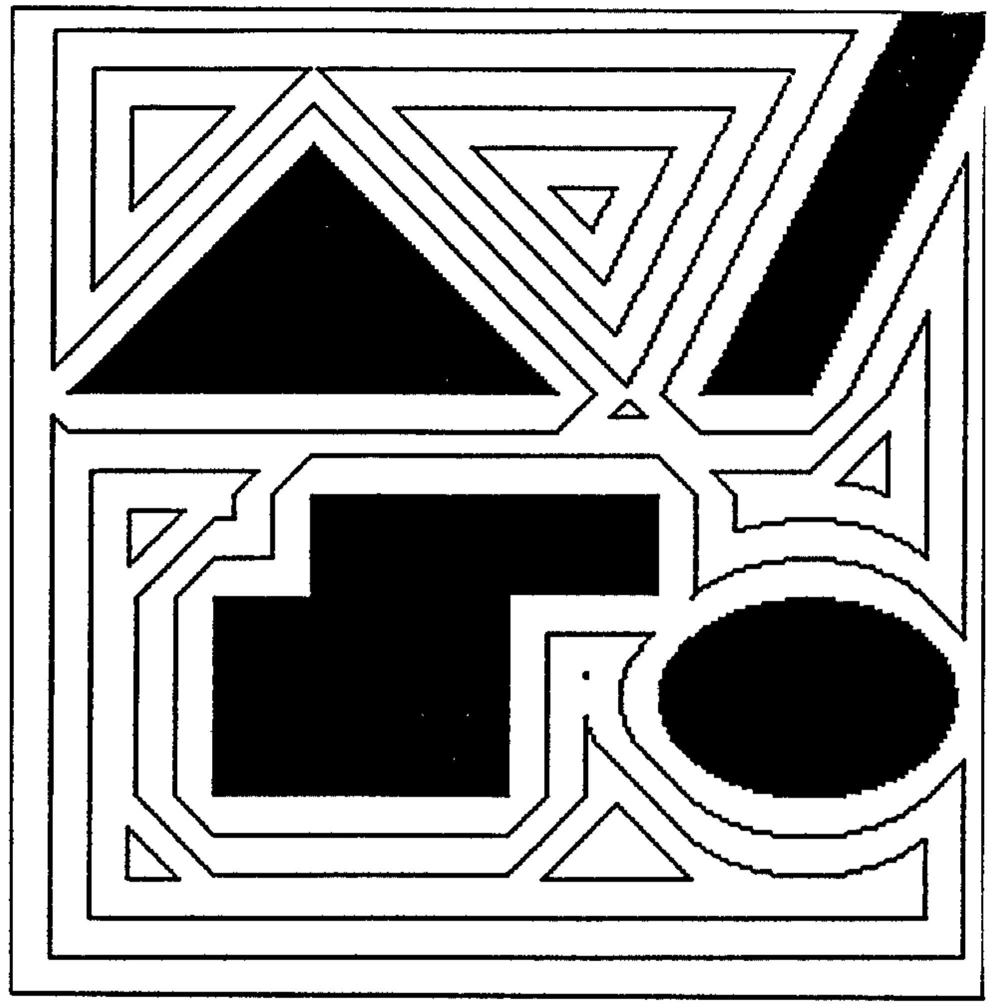
- obstacle configuration



[Barranquand, Langlois, Latombe, 1989]

Potential field approach

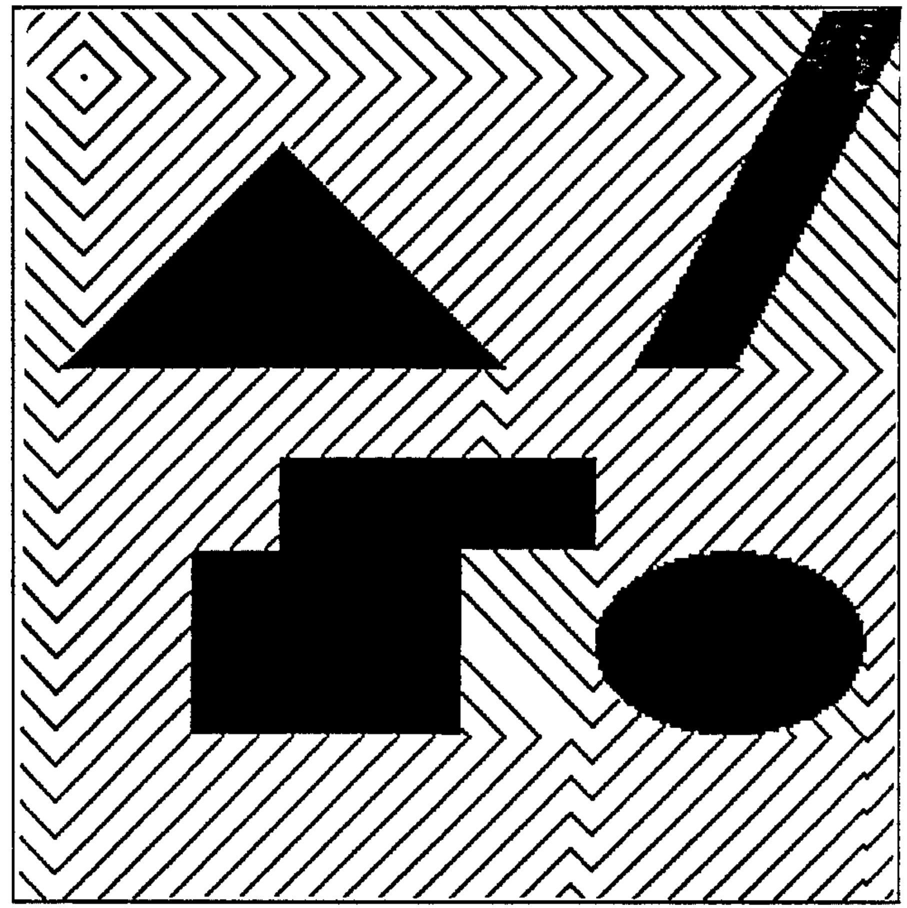
- contours of associated obstacle potential field



[Barranquand, Langlois, Latombe, 1989]

Potential field approach

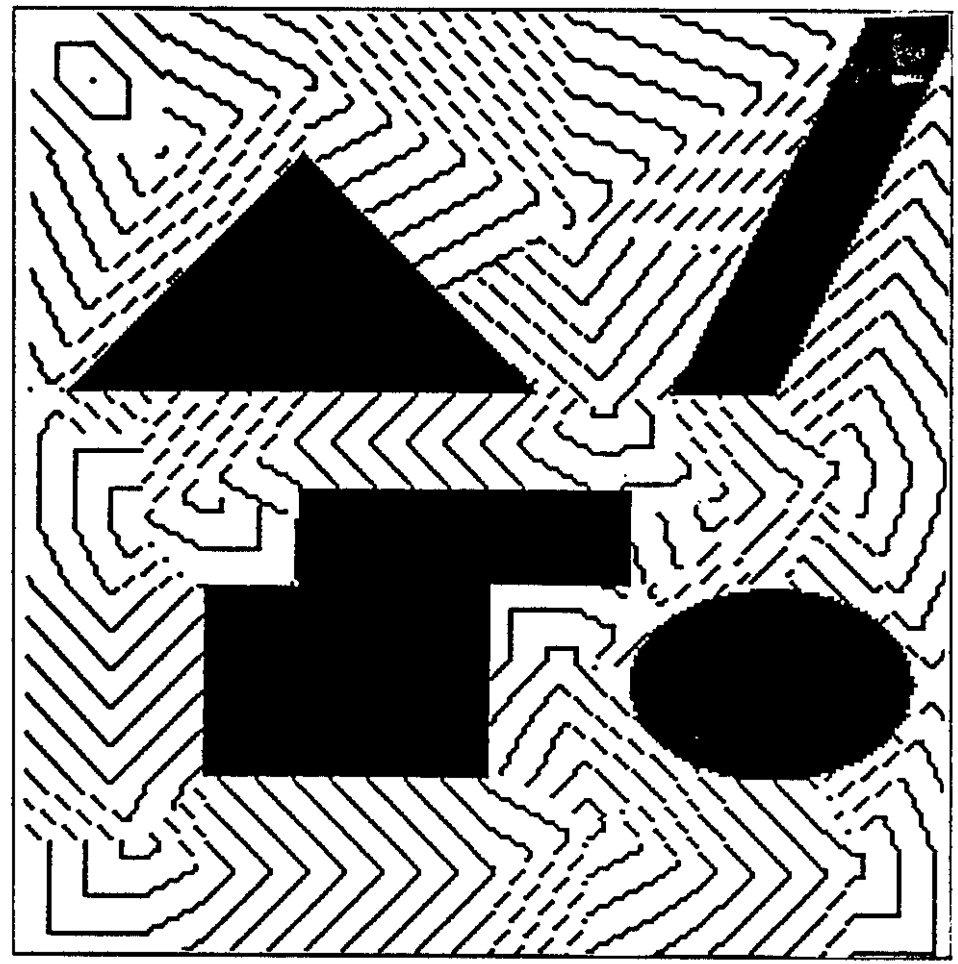
- contours of target potential field



[Barranquand, Langlois, Latombe, 1989]

Potential field approach

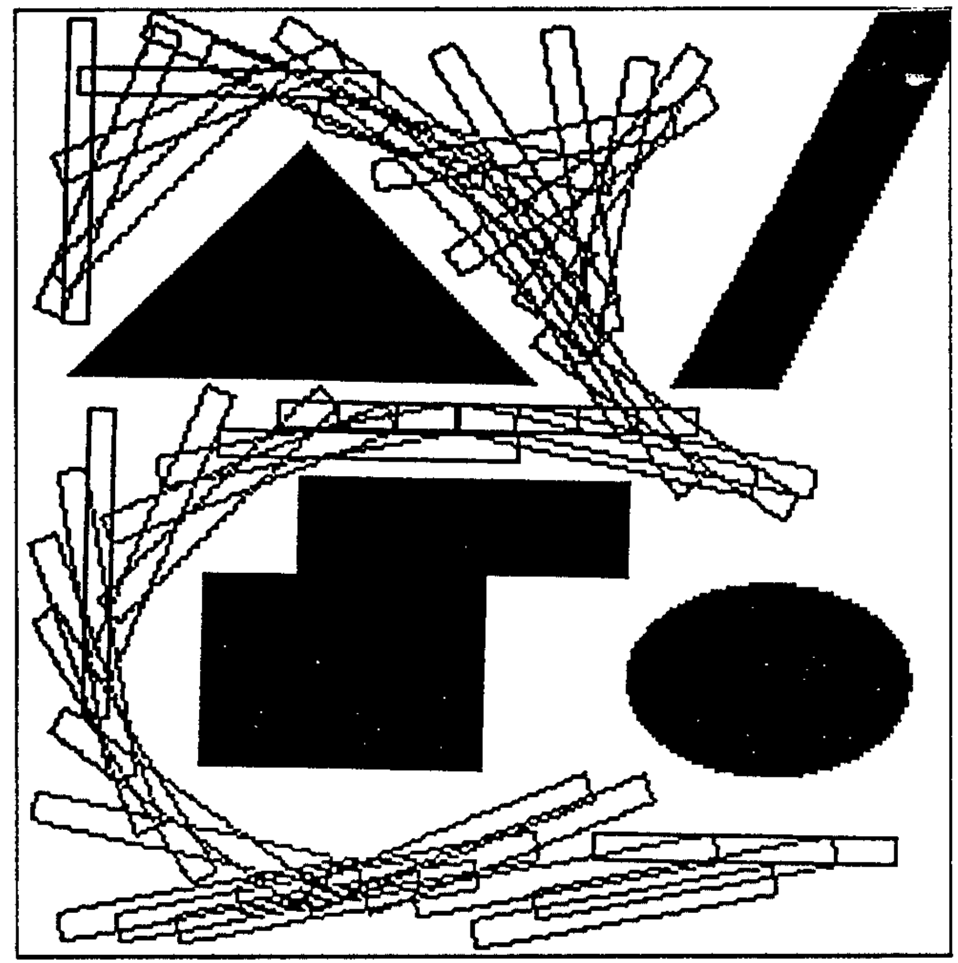
- contours of improved target potential field (by adding bubbles around obstacles)



[Barranquand, Langlois, Latombe, 1989]

Potential field approach

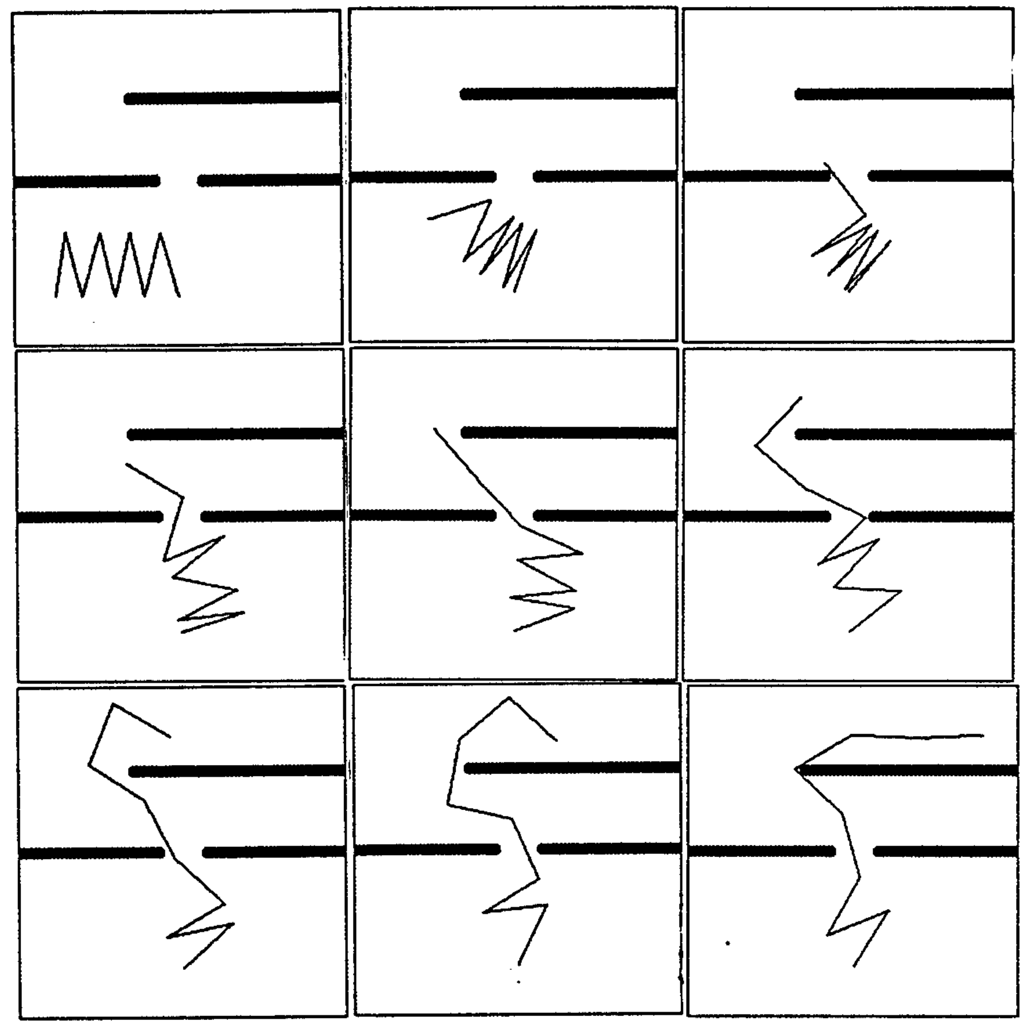
- adding all contributions leads to solution: gradient descent for vehicle



[Barranquand, Langlois, Latombe, 1989]

Potential field approach

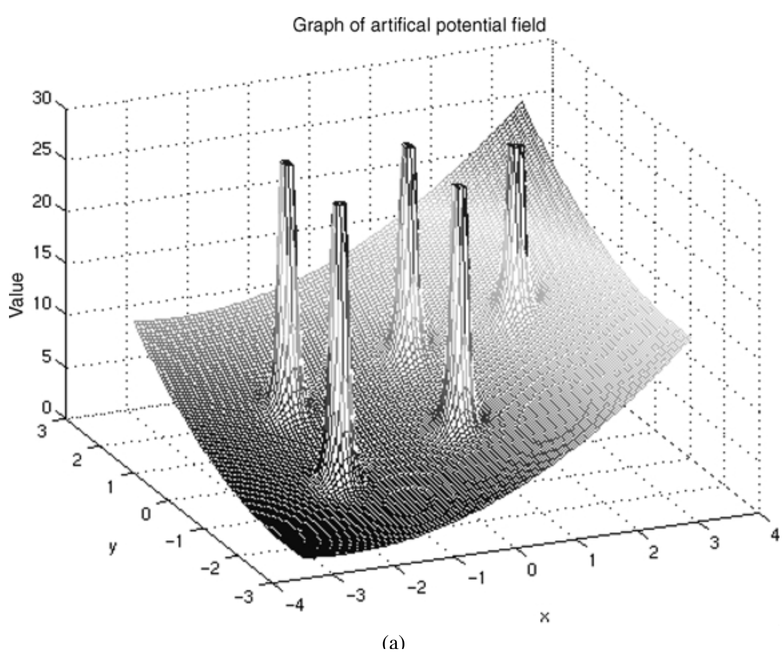
- generalization to higher-dimensional configuration spaces

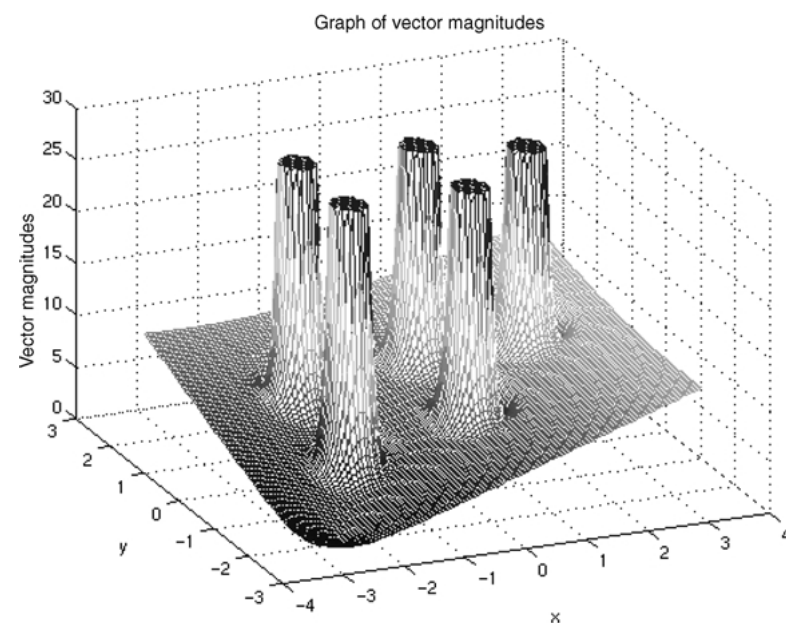


[Barranquand, Langlois, Latombe, 1989]

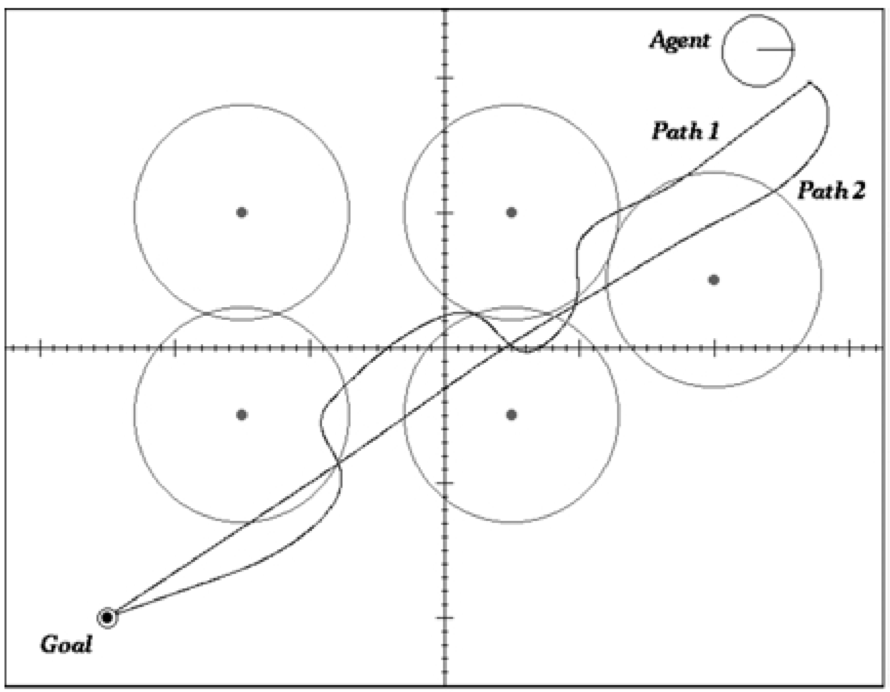
Comparison to human behavior

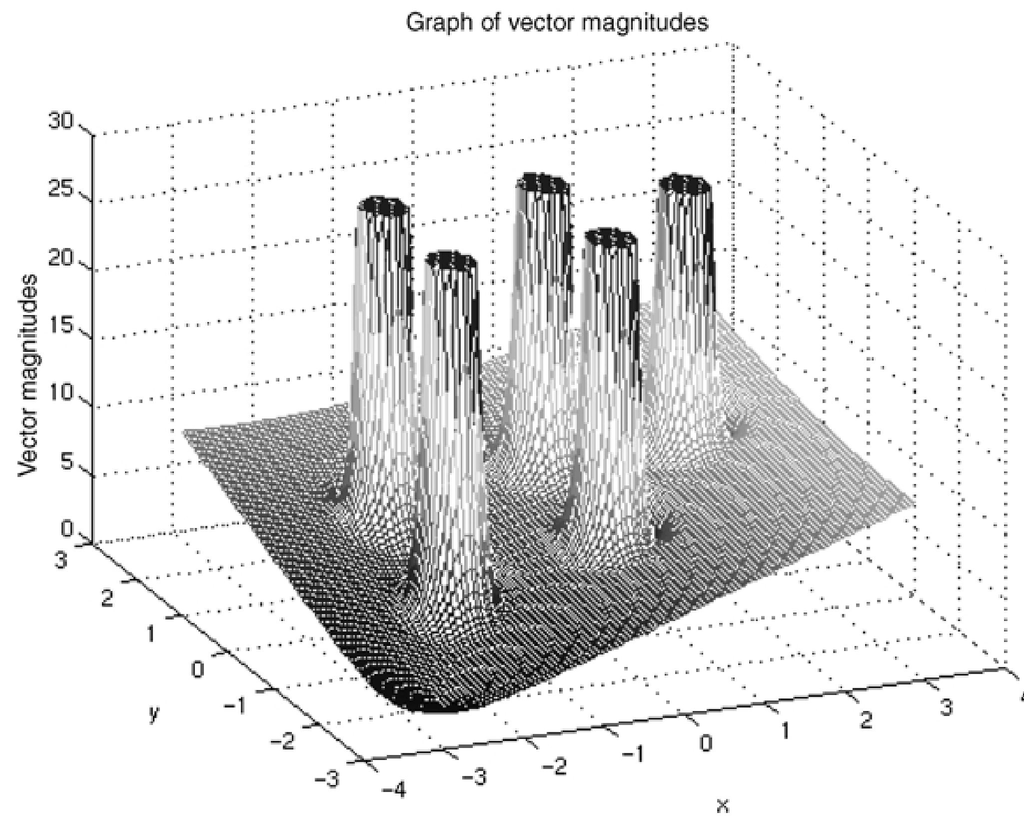
- Fajen/Warren compared the fit of a potential field approach to the fit of the attractor dynamics approach of human locomotion data





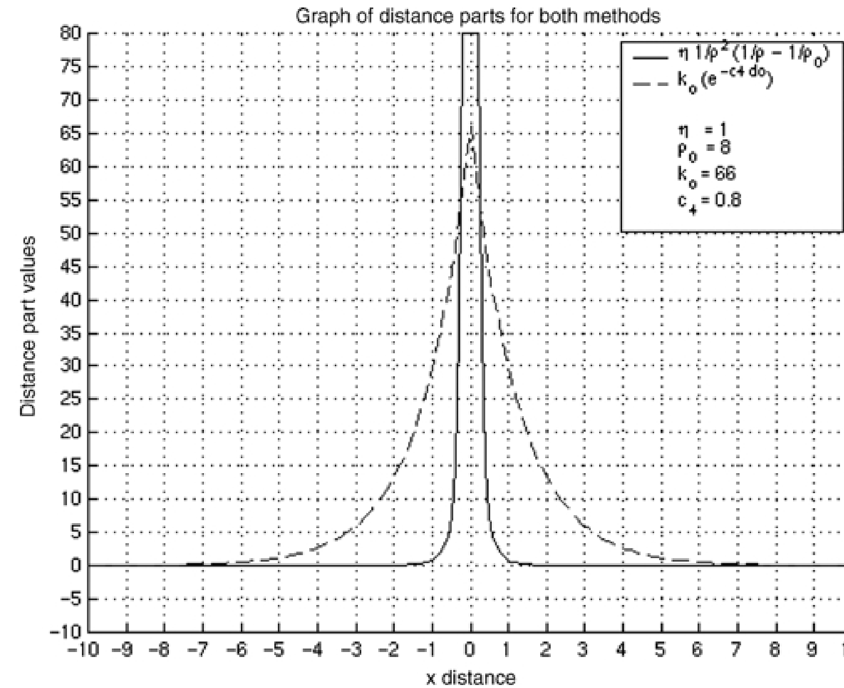
Comparison to human behavior



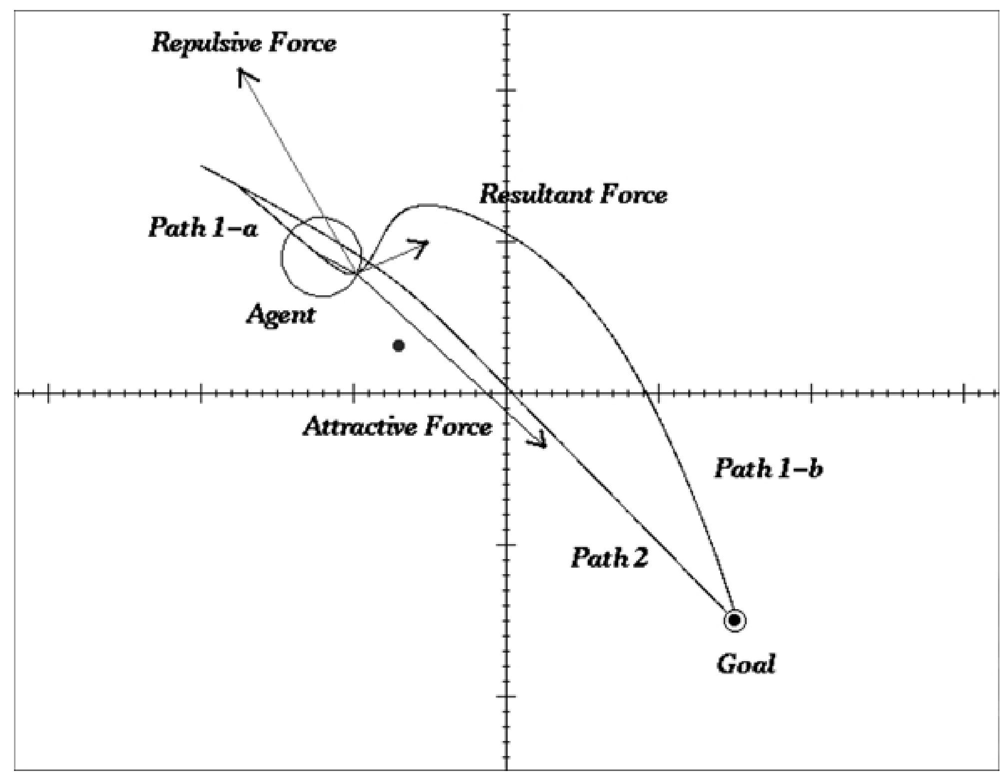


comparison potential field vs. attractor dynamics

- potential sharper than distance dependence of repellor



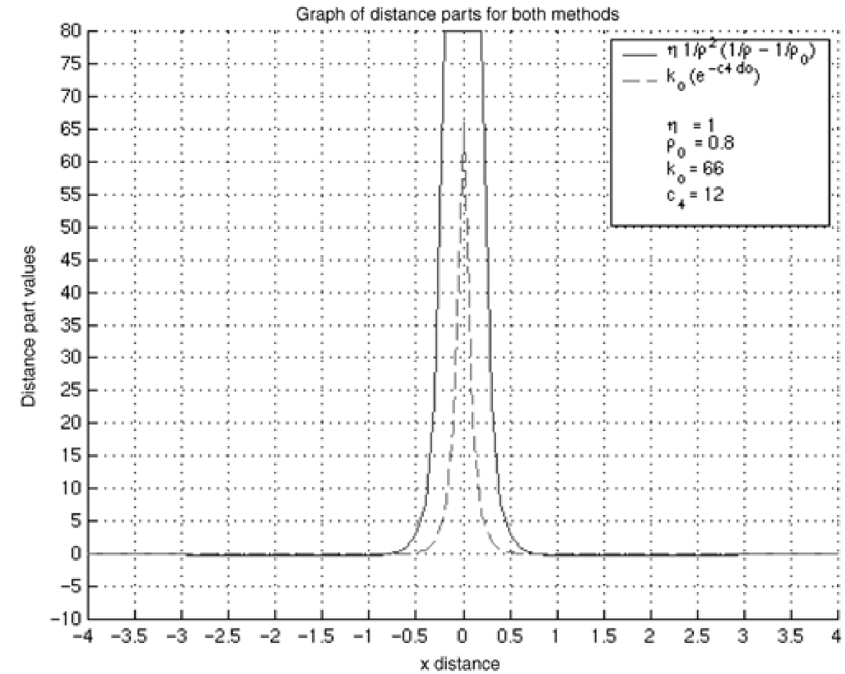
(a)



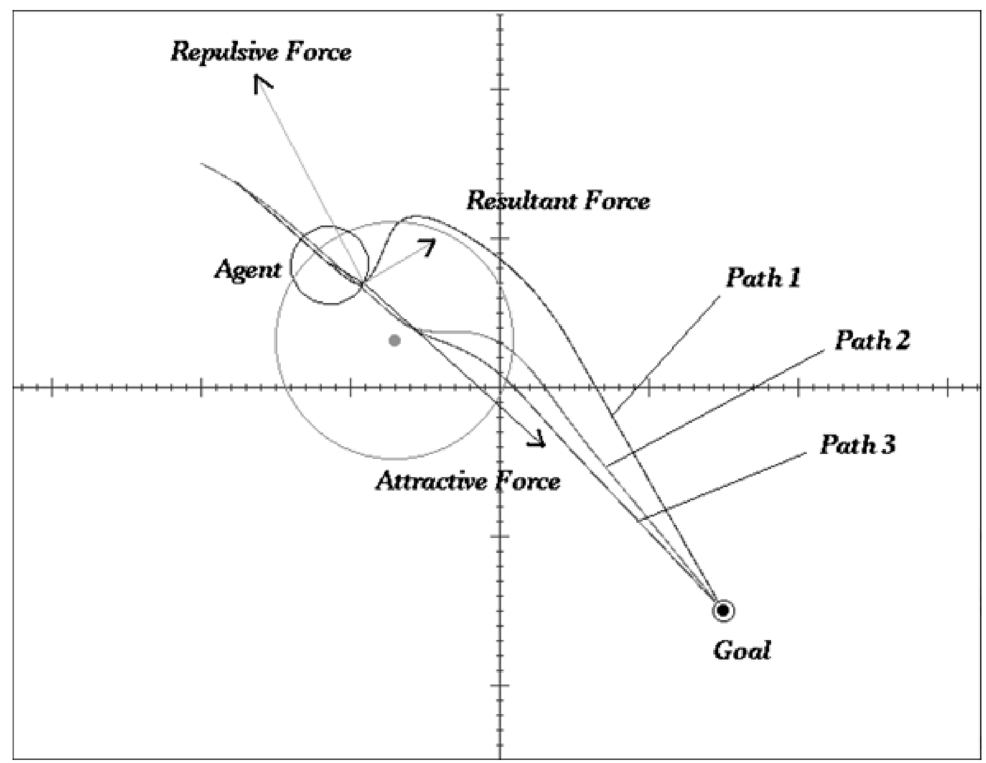
(b)

comparison potential field vs. attractor dynamics

- potential softer than distance dependence of repellor

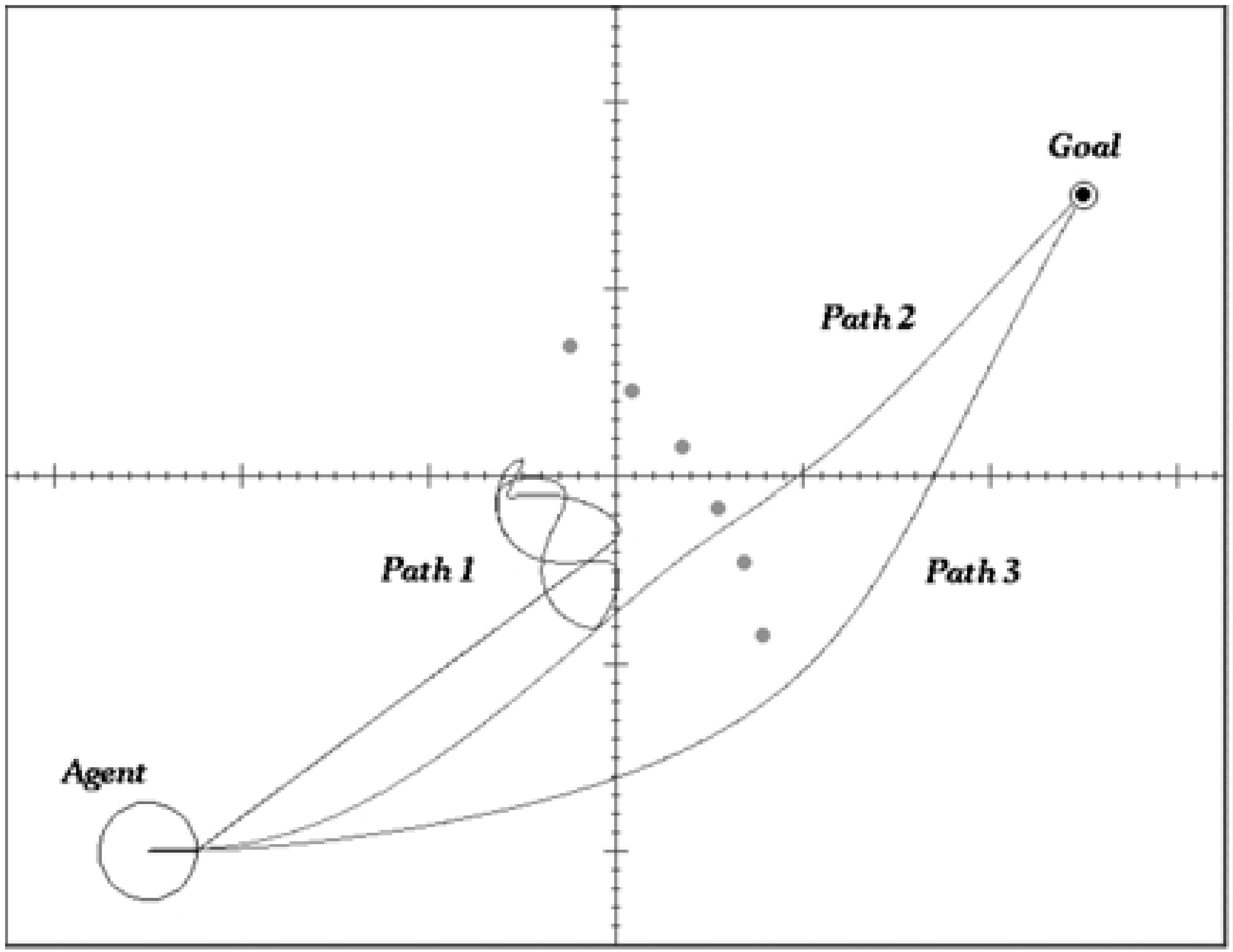


(a)



(b)

spurious attractors in potential field approach



Comments relative to attractor dynamics approach

- the problem of spurious attractors in AD: solution proposed in Dose, Schöner: reduce number of contributions to avoid cancelation
- the problem obstacle width: that concept actually exists… as you saw in the exercises…

Potential fields: limitations

- spurious attractors and constraint violations
- solution: making potential field approach exact and global: navigation functions
- potential computed such that it only has the right maxima and minimal
- but: computational cost
- but: requires global information

Extension of attractor dynamics approach

Autonomous Robots (2019) 43:589–610
https://doi.org/10.1007/s10514-018-9729-2

CrossMark

Attractor dynamics approach to joint transportation by autonomous robots: theory, implementation and validation on the factory floor

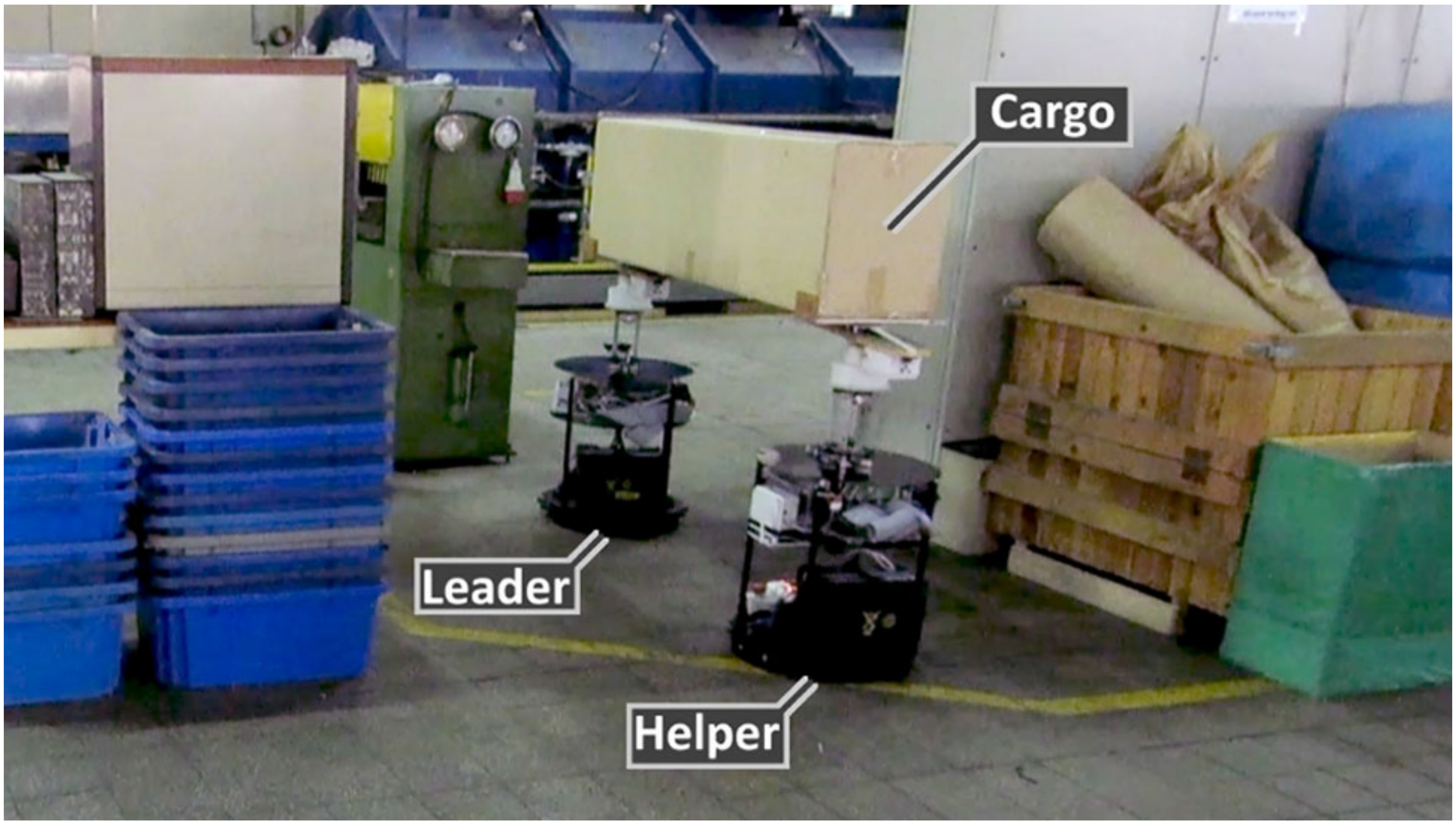
Toni Machado[1] · Tiago Malheiro[1] · Sérgio Monteiro[1] · Wolfram Erlhagen[2] · Estela Bicho[1]

Received: 1 November 2016 / Accepted: 2 April 2018 / Published online: 12 April 2018

Abstract

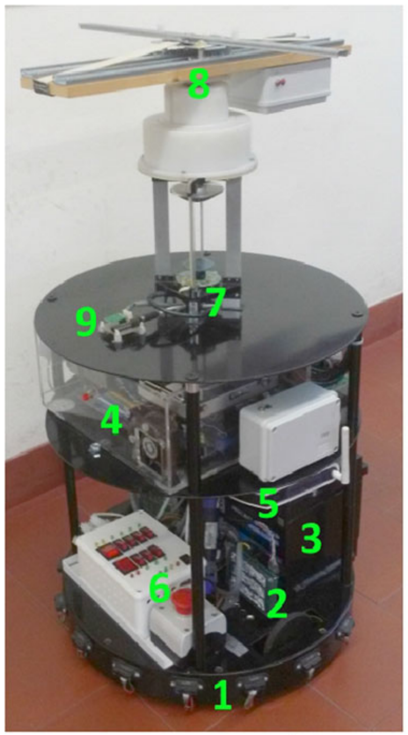
This paper shows how *non-linear attractor dynamics* can be used to control teams of two autonomous mobile robots that coordinate their motion in order to transport large payloads in unknown environments, which might change over time and may include narrow passages, corners and sharp U-turns. Each robot generates its collision-free motion online as the sensed information changes. The control architecture for each robot is formalized as a non-linear dynamical system, where by design attractor states, i.e. asymptotically stable states, dominate and evolve over time. Implementation details are provided, and it is further shown that odometry or calibration errors are of no significance. Results demonstrate flexible and stable behavior in different circumstances: when the payload is of different sizes; when the layout of the environment changes from one run to another; when the environment is dynamic—e.g. following moving targets and avoiding moving obstacles; and when abrupt disturbances challenge team behavior during the execution of the joint transportation task.

Extension of attractor dynamics approach



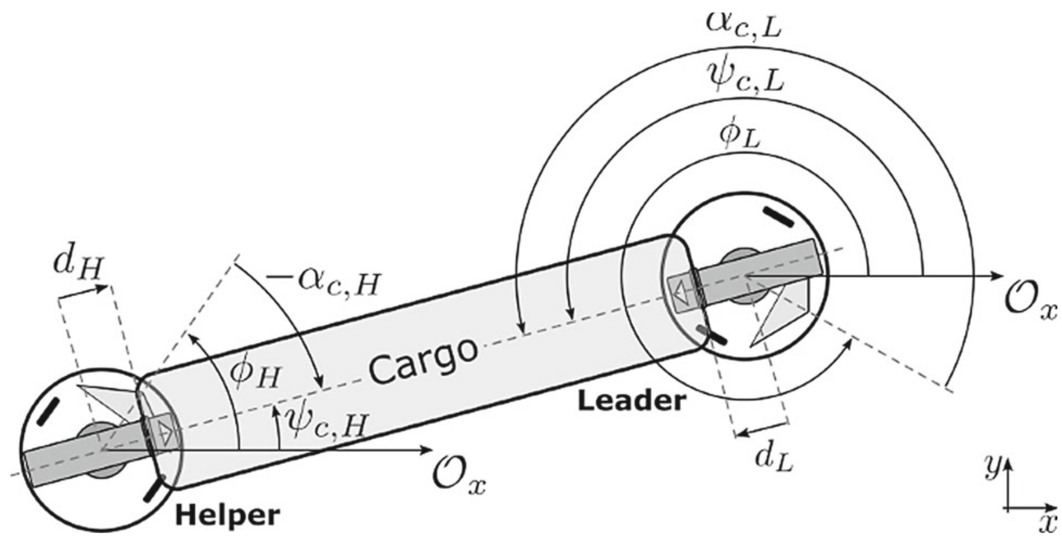
[Machado et al, 2019]

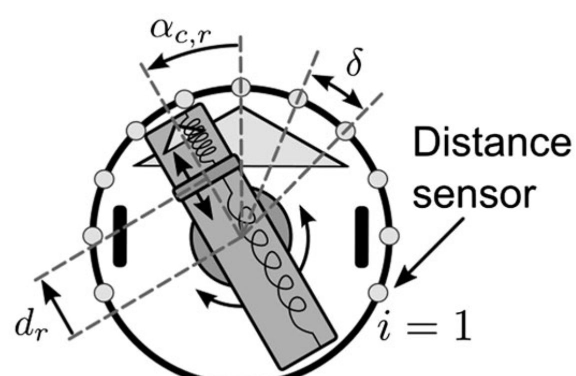
Extension of attractor dynamics approach



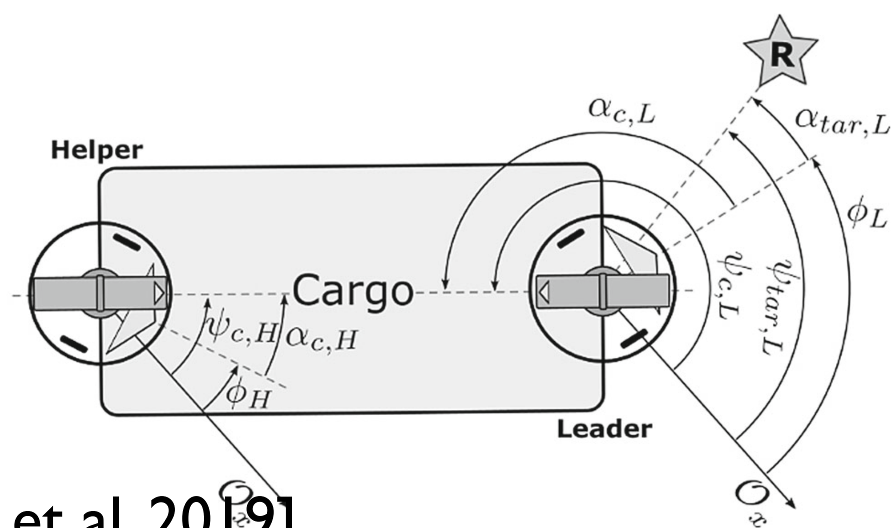
1. Distance sensor
2. Motorized wheel
3. Battery
4. Computer
5. Wireless router
6. Power adapter
7. Vision System
8. Support 2 DoF
9. Compass

(a)



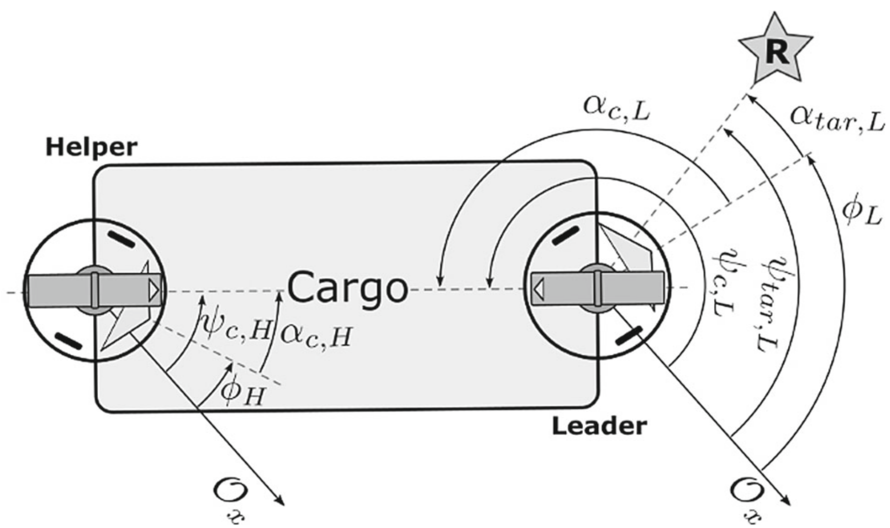


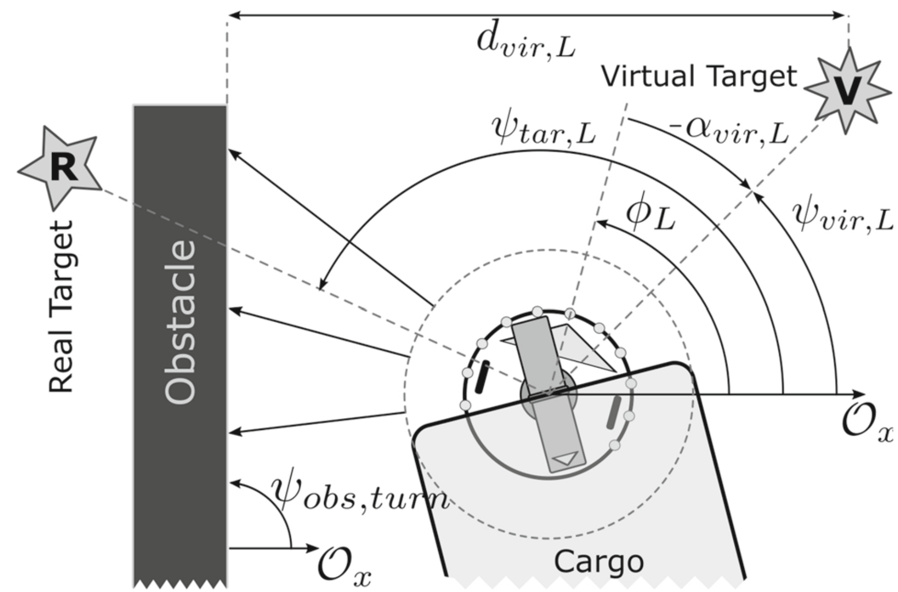
(b)

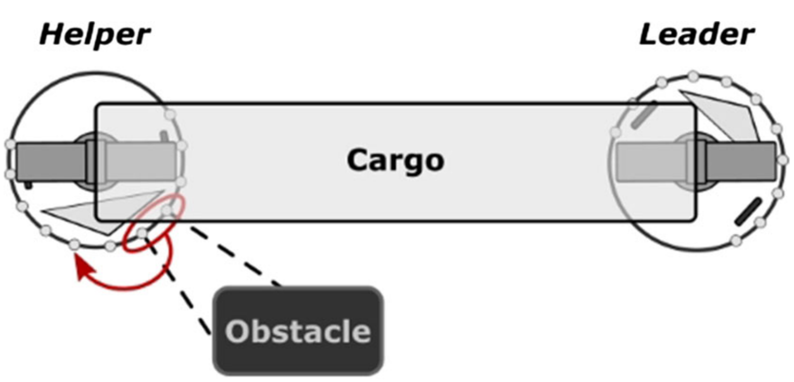


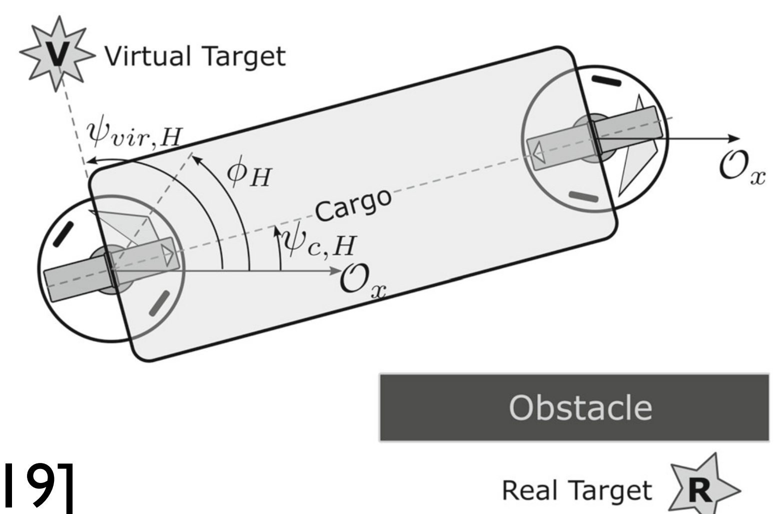
[Machado et al, 2019]

Extension of attractor dynamics approach



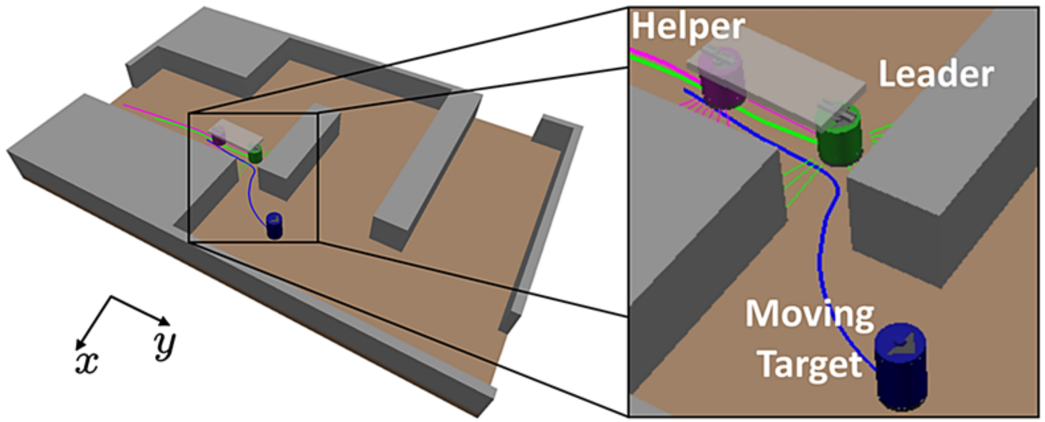




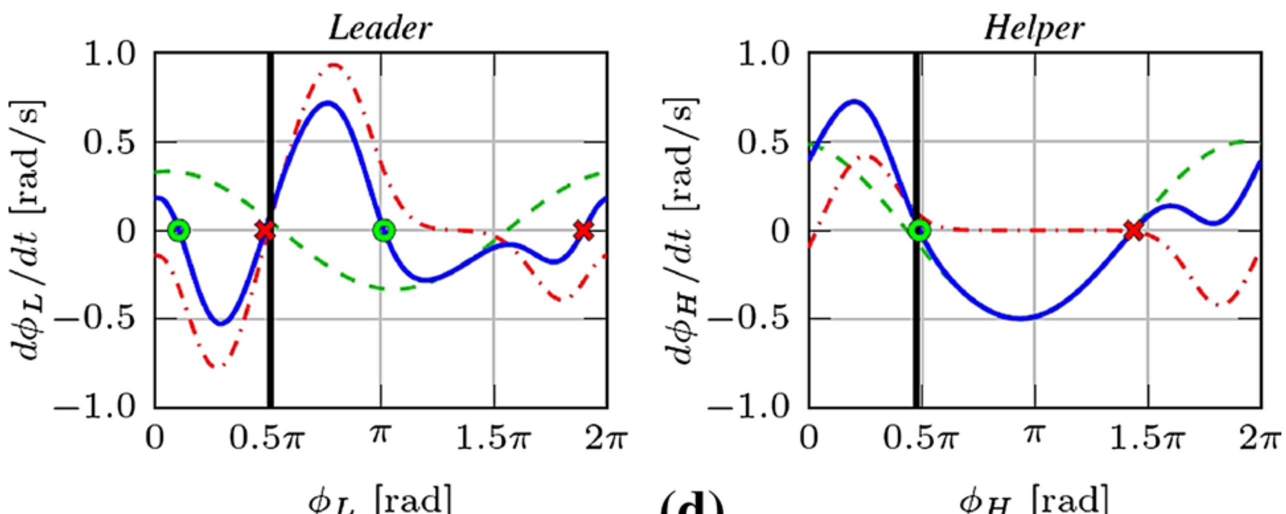


[Machado et al, 2019]

Extension of attractor dynamics approach



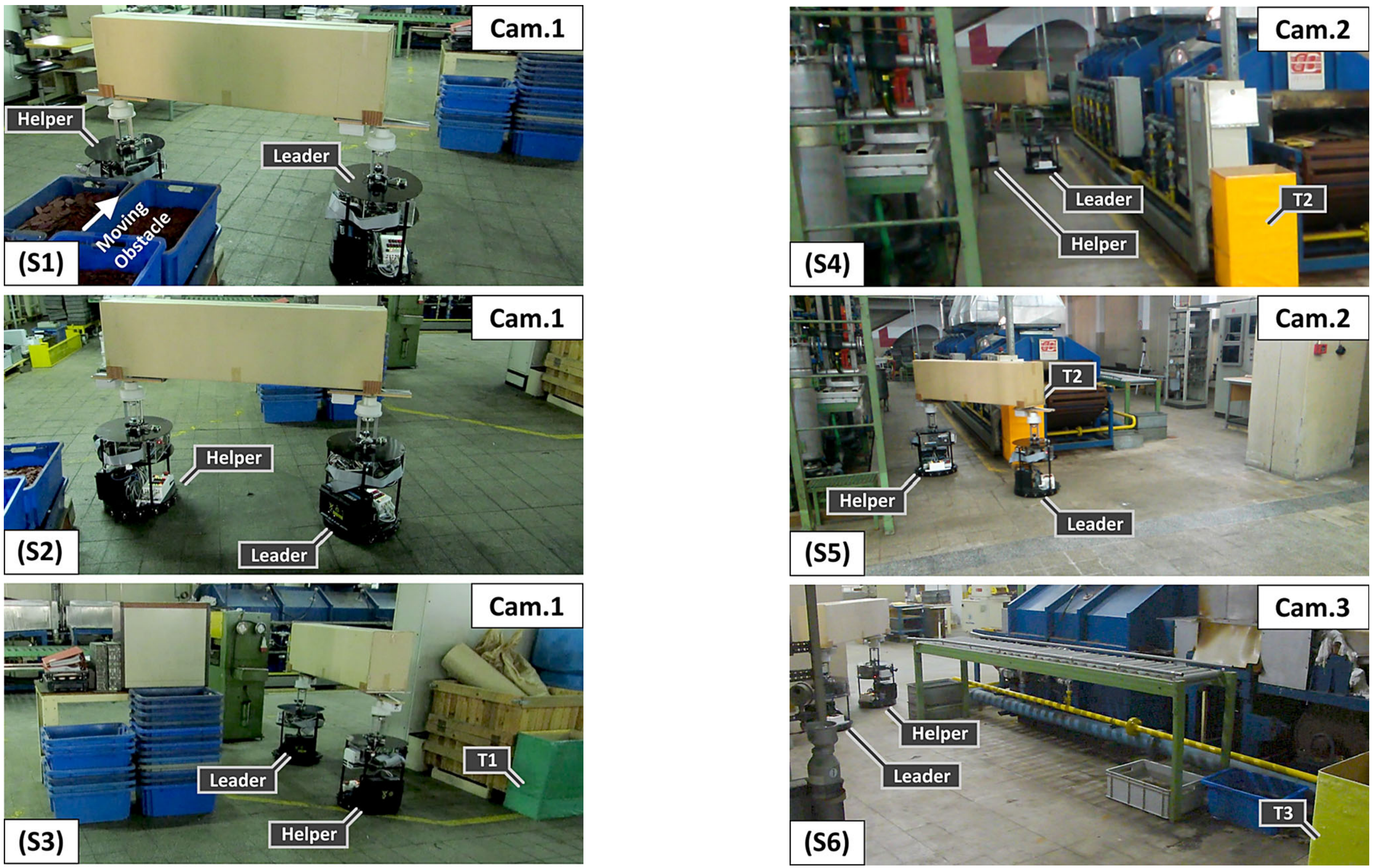
(c)



(d)

[Machado et al, 2019]

Extension of attractor dynamics approach

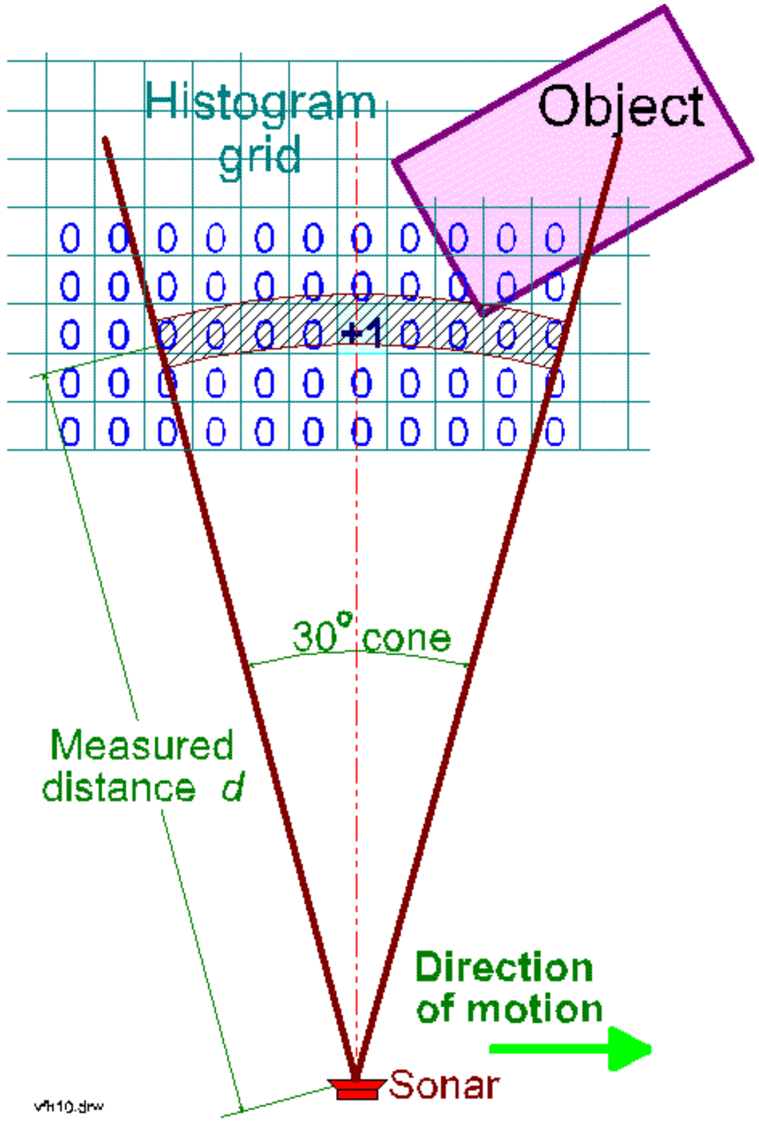


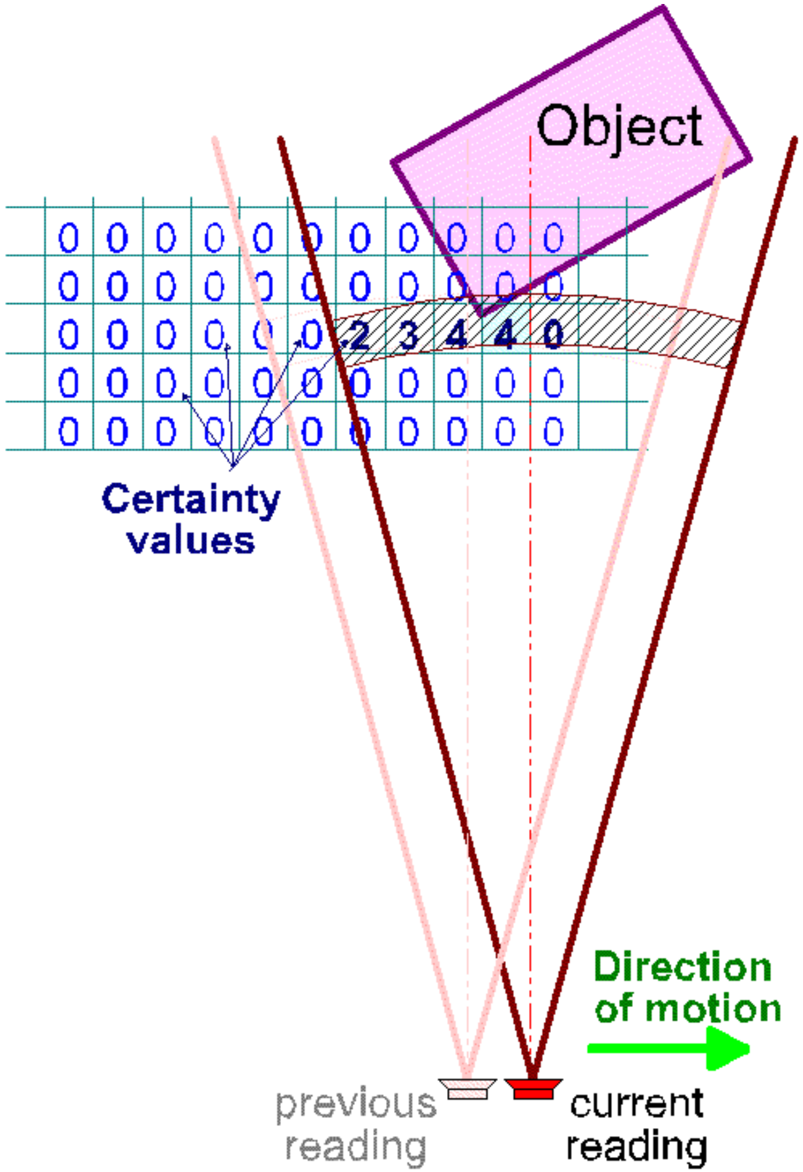
[Machado et al, 2019]

Virtual force field: Borenstein & Koren

- ultra-sound histograms: the virtual force field concept
- vector-field histogram concept: polar histogram (heading direction!); height (strength) depends on both certainty and distance
- threshold: determine free sectors
- select free direction closest to target

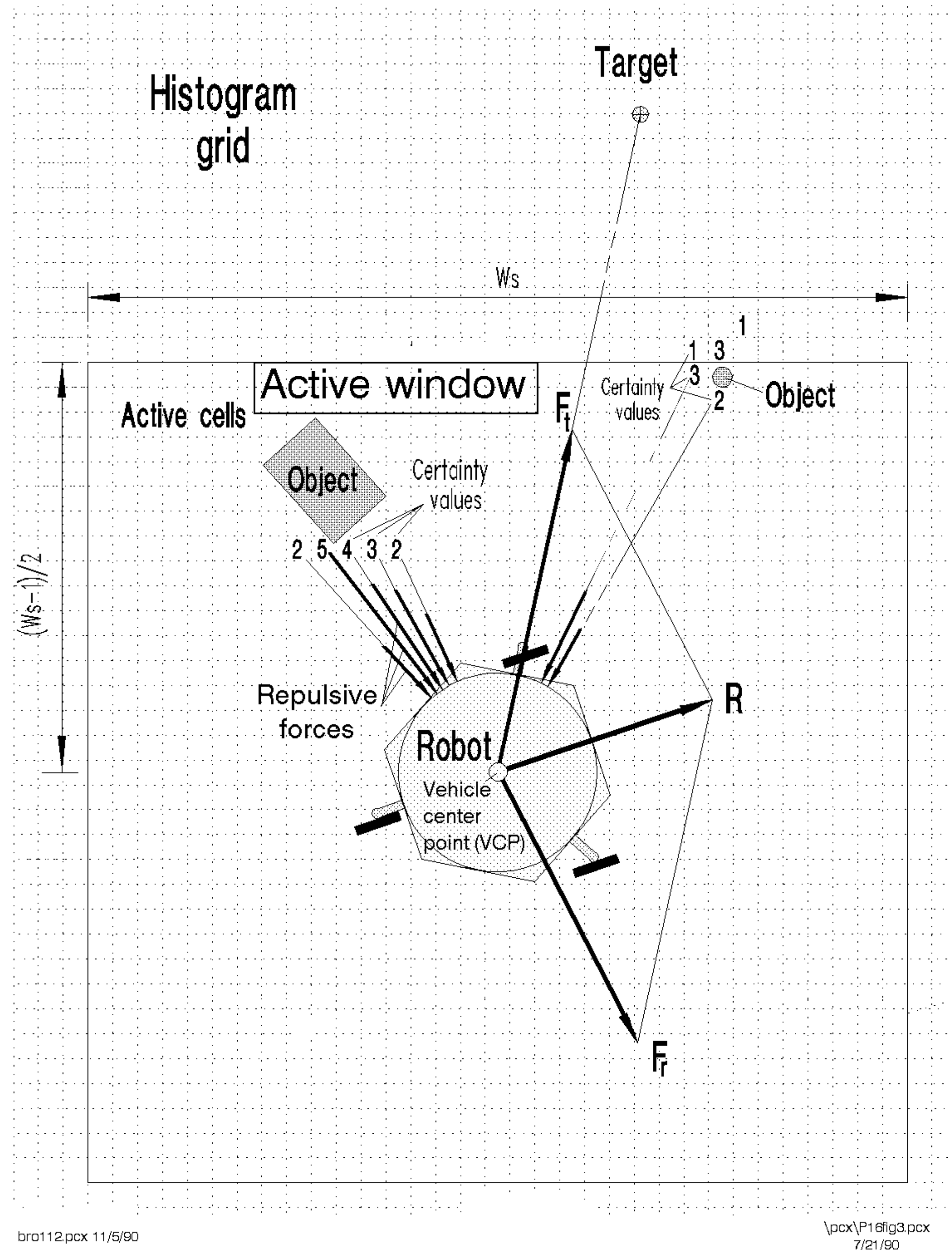
Virtual force field: Borenstein & Koren





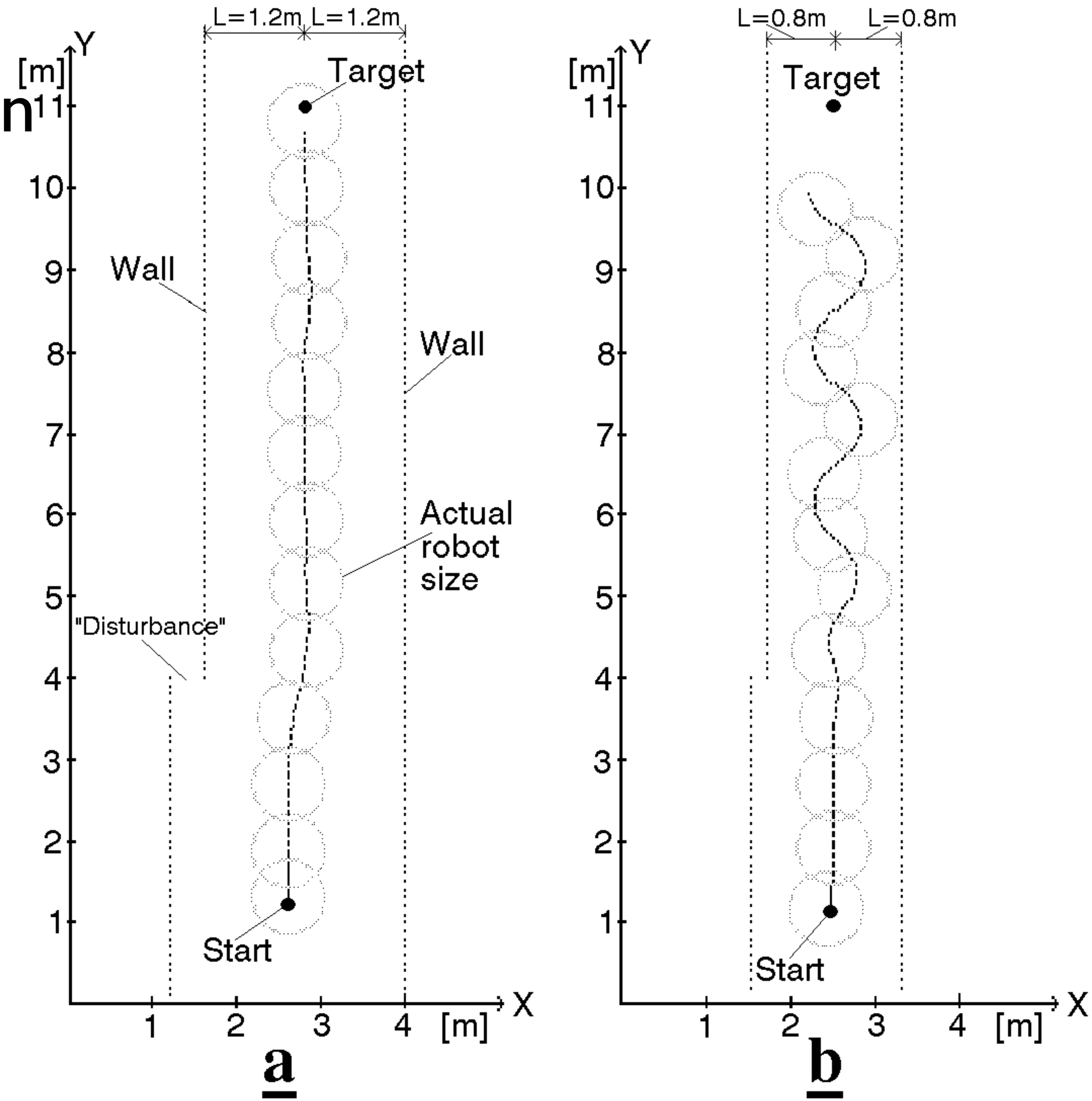
Virtual force field: Borenstein & Koren

- vector toward target
- active window around robot
- use histogram within active window to compute vectors pointing away from obstacle
- vector summing
- ~dynamic approach!



Virtual force field: Borenstein & Koren

- Problem: oscillations in narrow passages



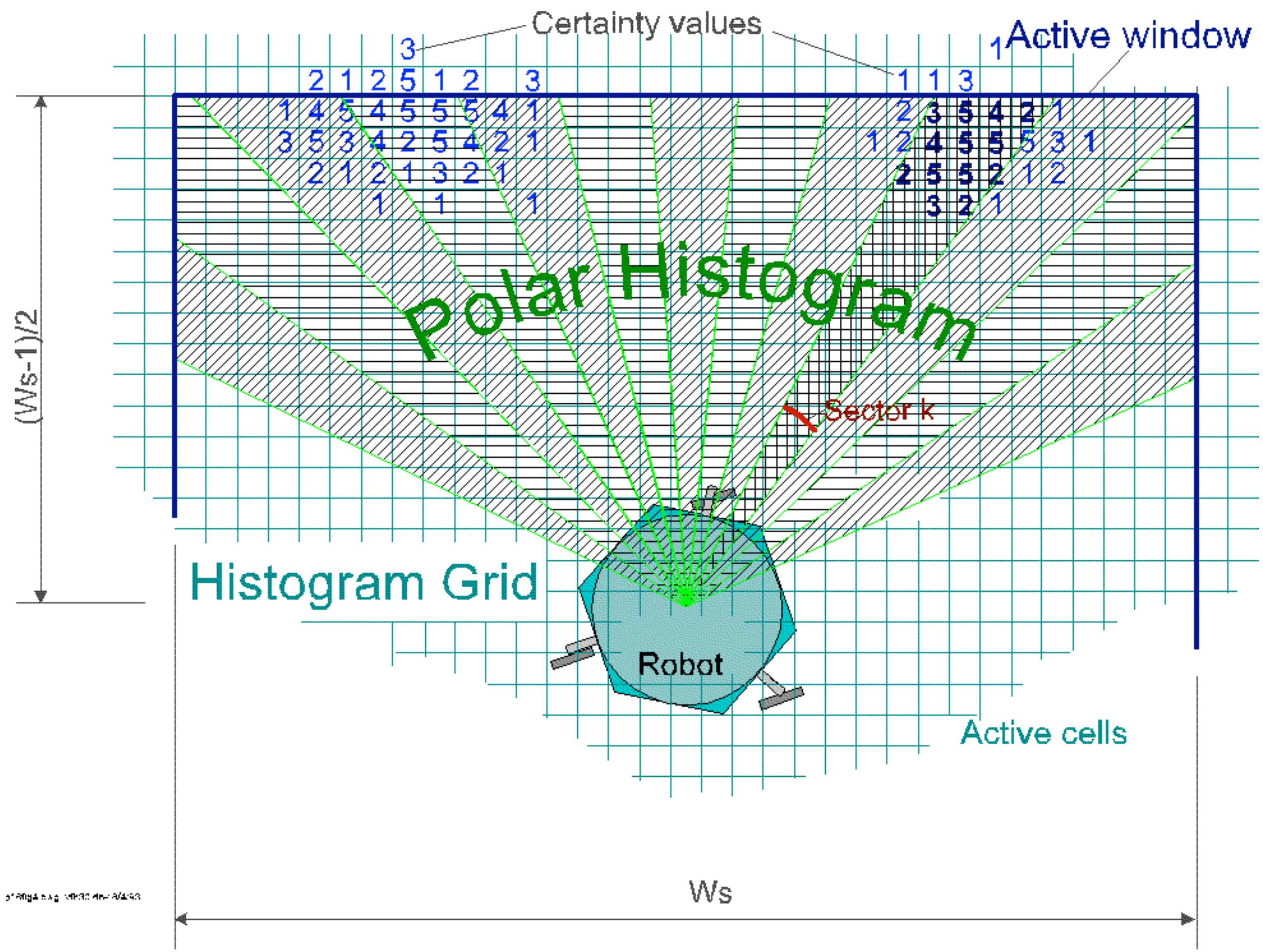
Stable motion in wide corridor V=0.8m/s

Unstable motion in narrow corridor. V=0.8m/sec.

bro113.pcx 7/27/90

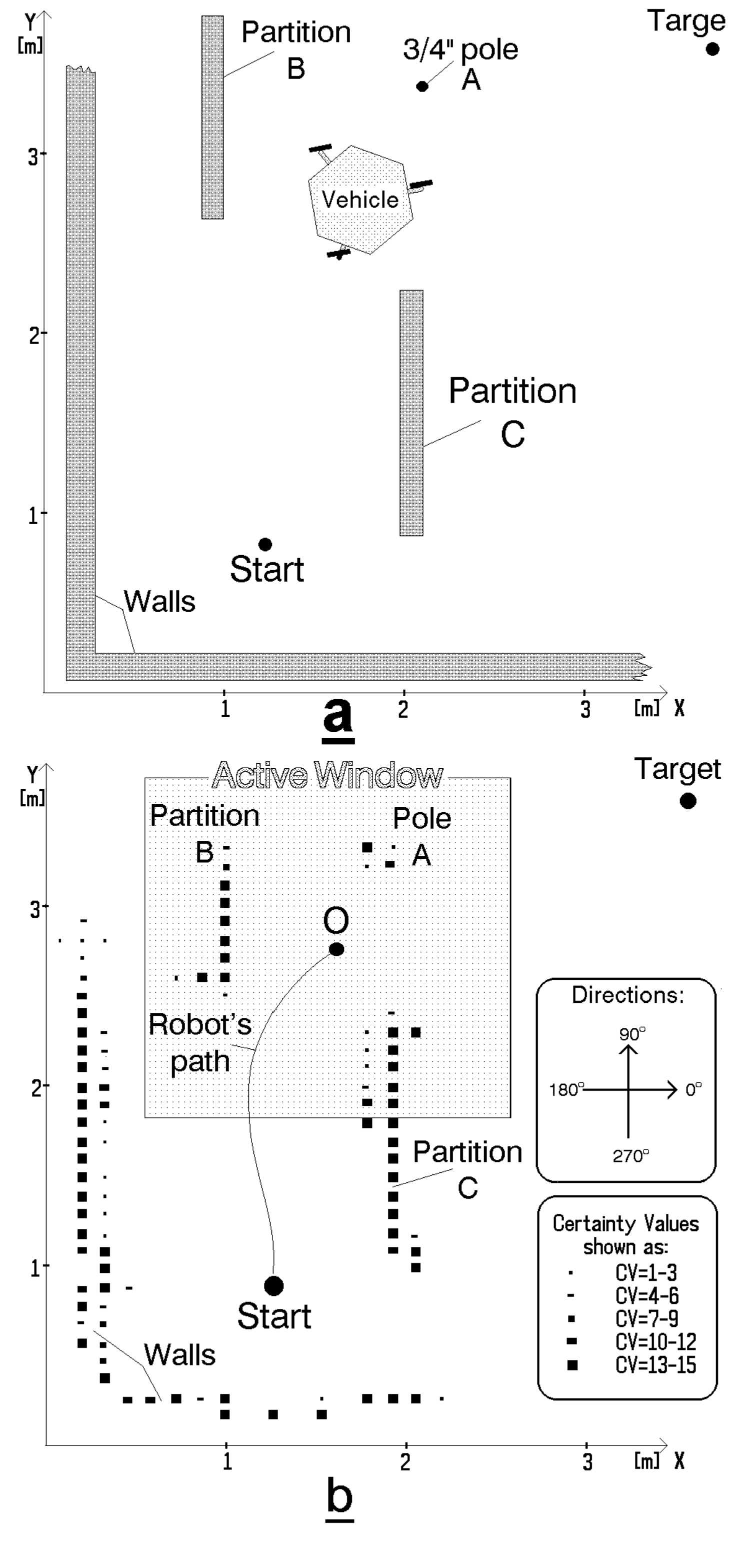
Vector field histogram: Borenstein & Koren

- transform active window in world grid into polar histogram



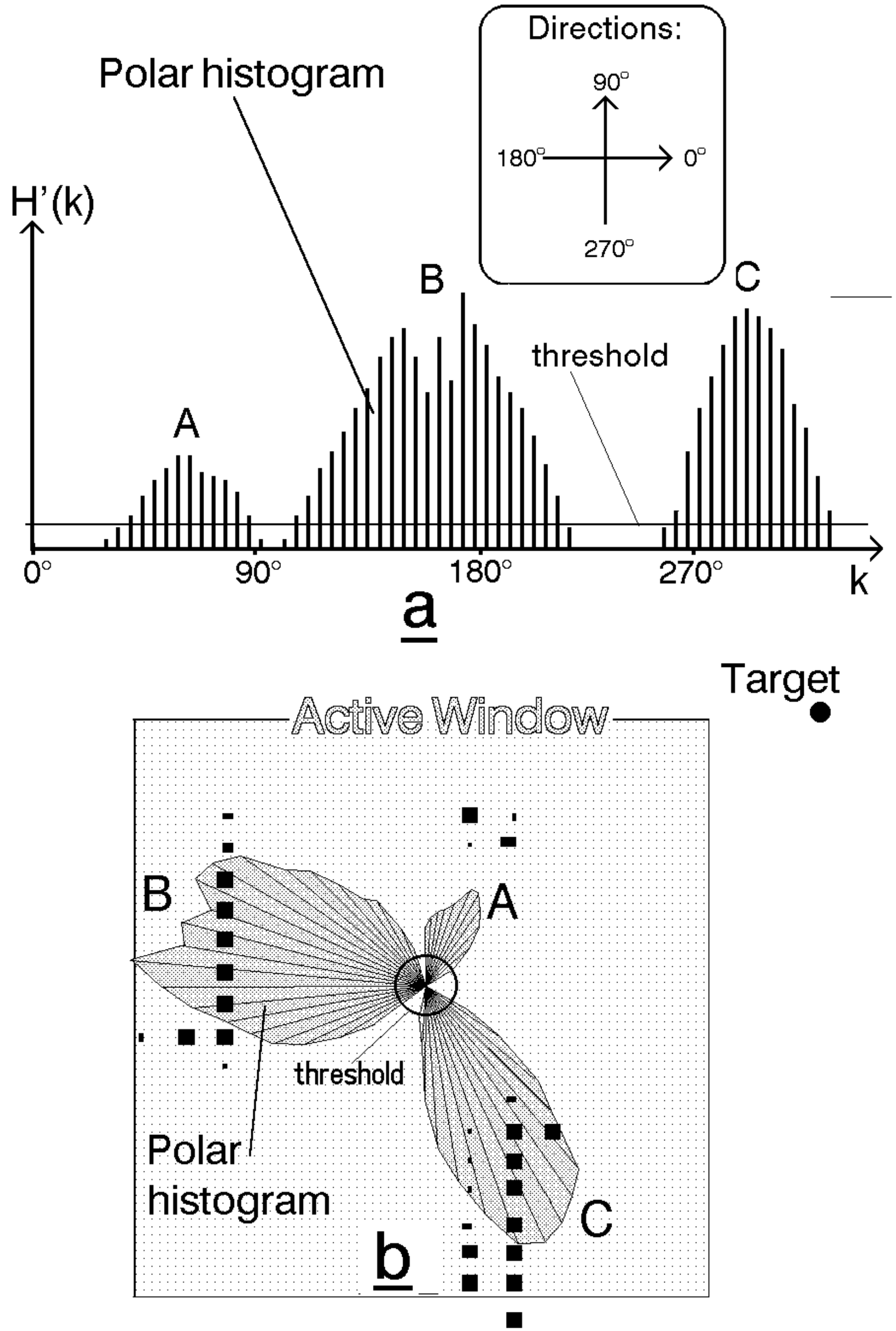
Vector field histogram: Borenstein & Koren

- lab set-up



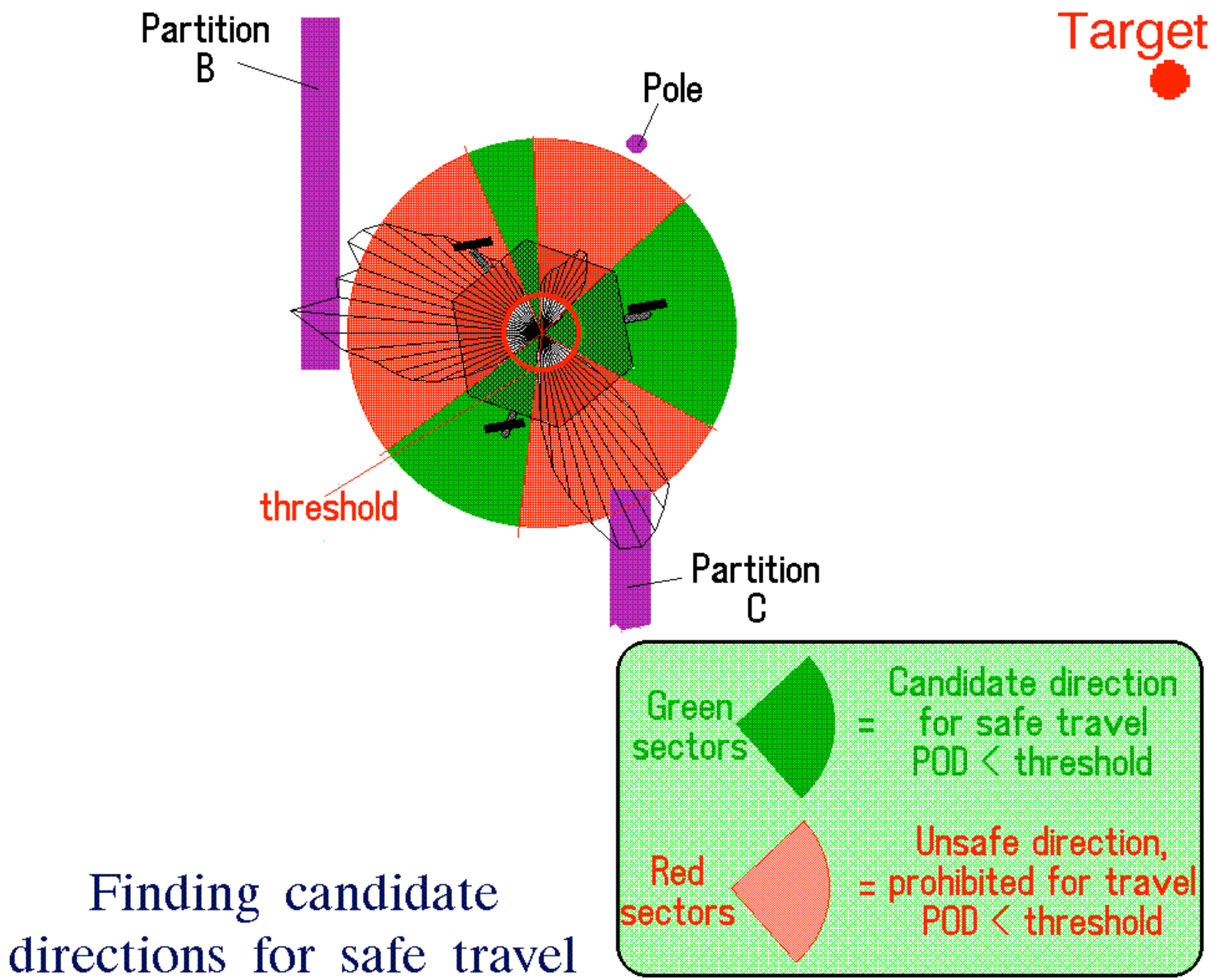
Vector field histogram: Borenstein & Koren

- local polar histogram provides "free" directions



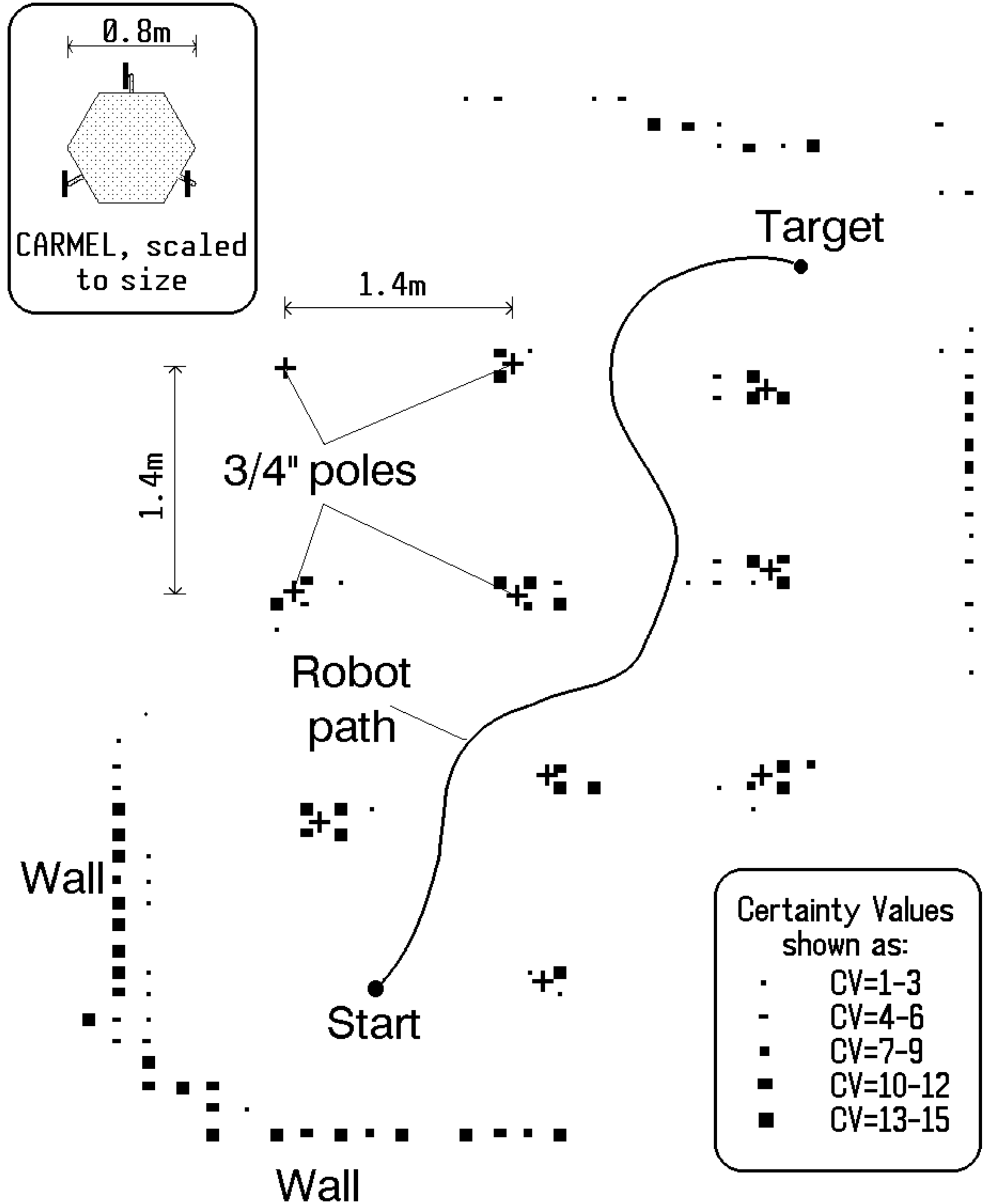
Vector field histogram: Borenstein & Koren

- Select safe direction algorithmical



Vector field histogram: Borenstein & Koren

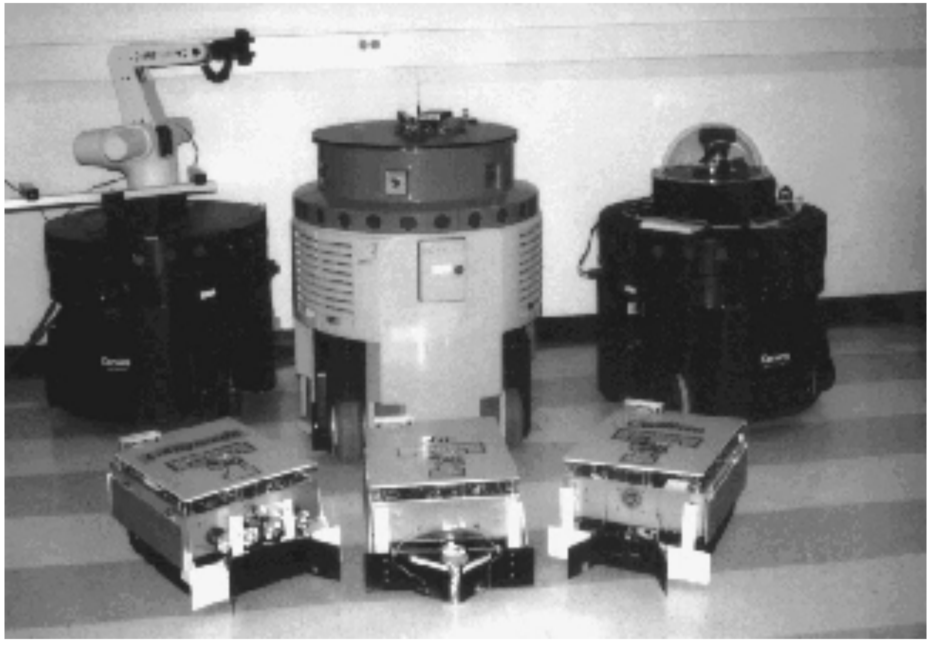
- works



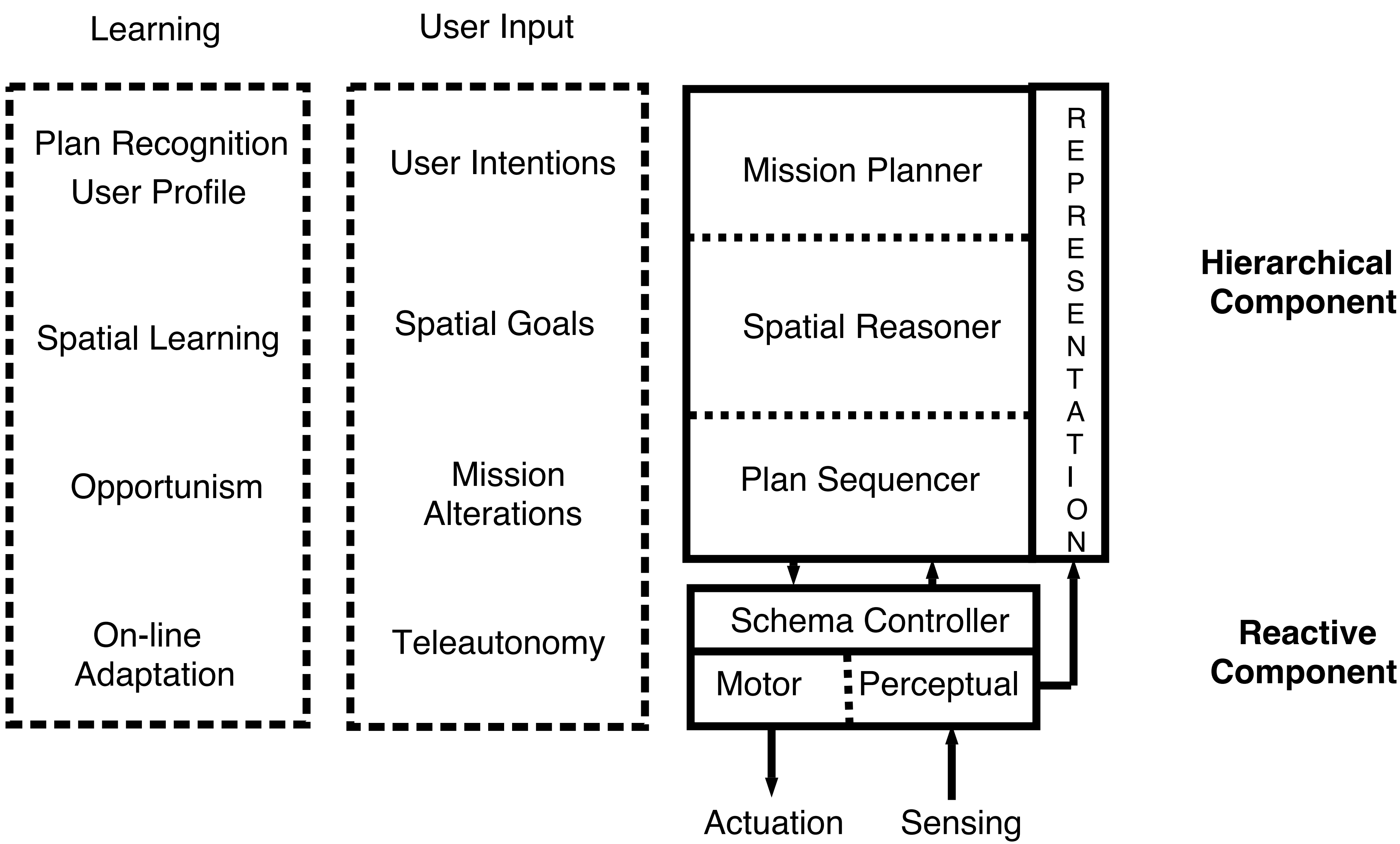
Potential fields as reactive planners

- use potential field to plan locally based on low-level sensory information (reactive)
- different "behaviors" generated by different vector-fields ("schema", slight generalization of potential fields)
- organize the different behaviors in an architecture

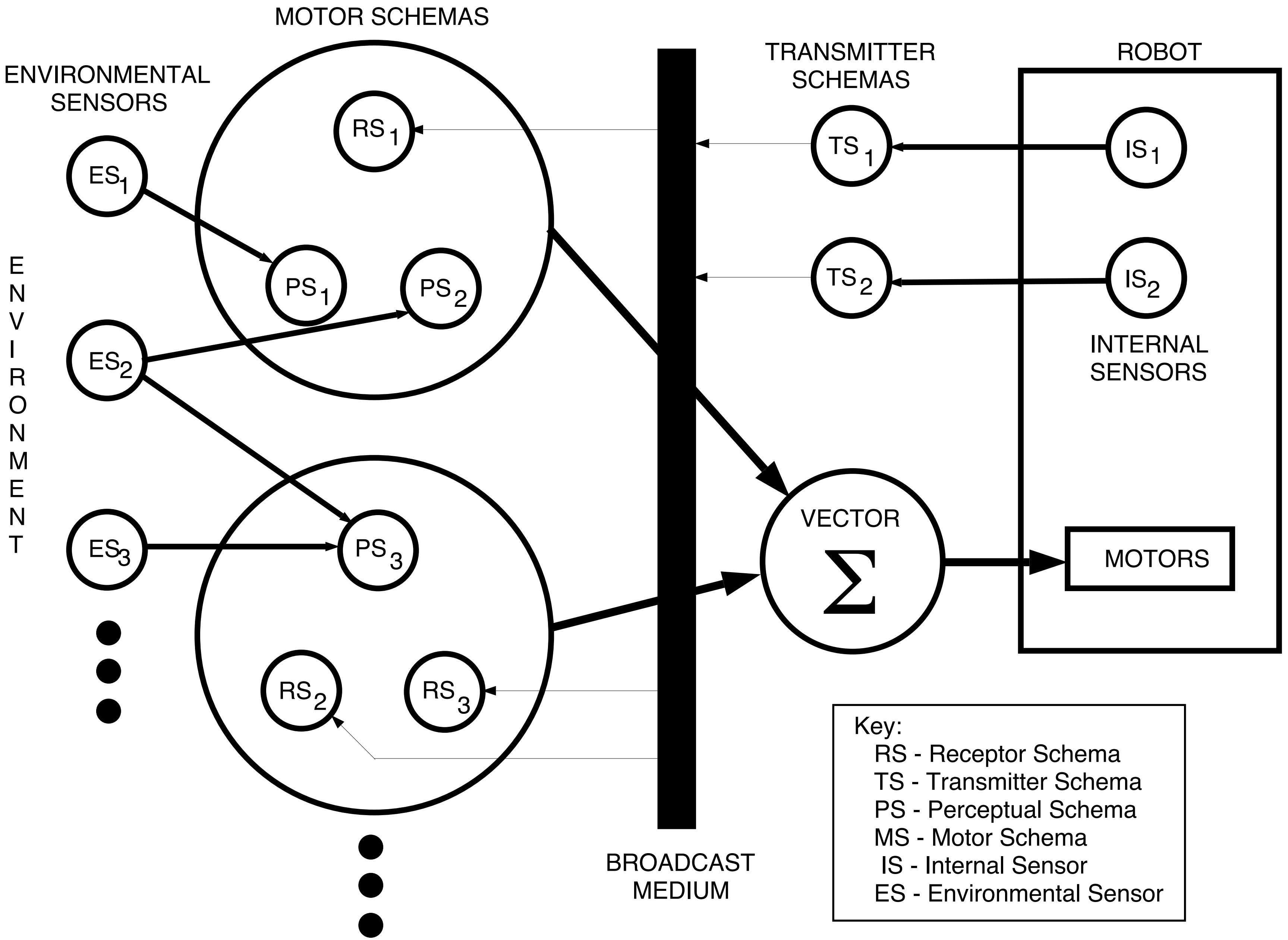
[Arkin, Blach: AuRA 1997]



Architecture



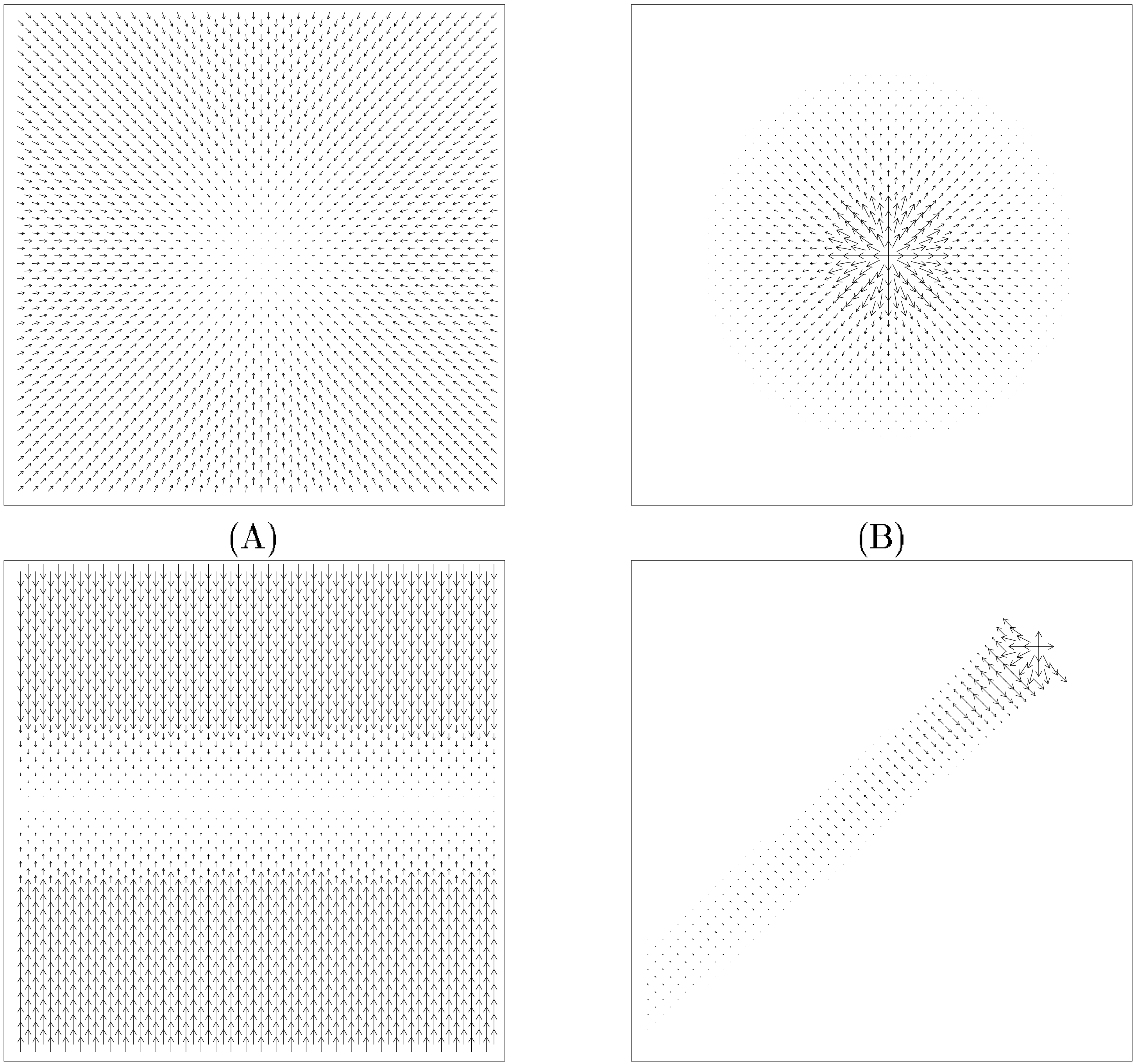
The reactive component



Motor schemata

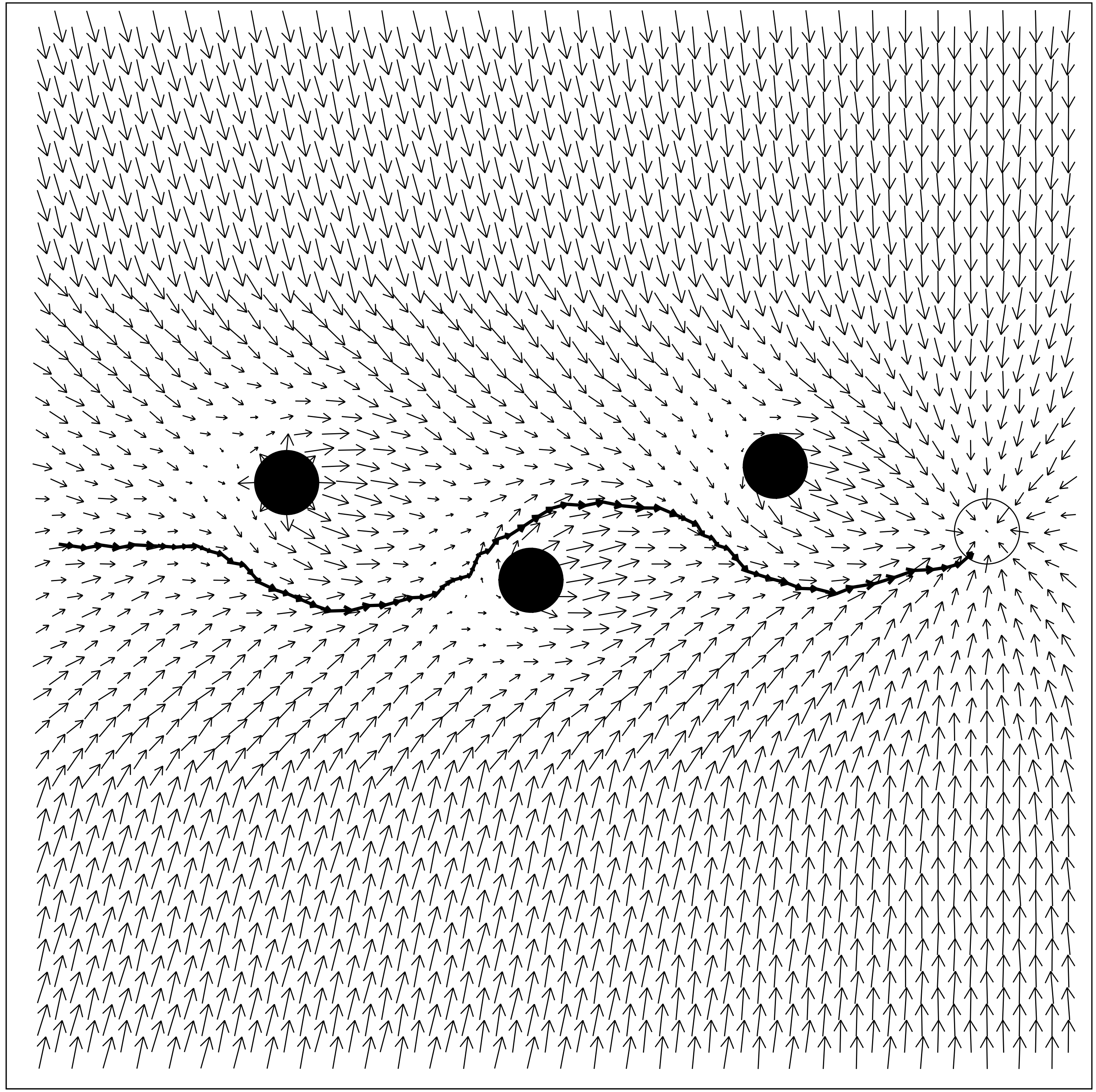
- **Move-ahead**: move in a particular compass direction.
- **Move-to-goal** (both ballistic and guarded): move towards a discrete stimulus.
- **Stay-on-path**: move towards the center of a discernible pathway, e.g., a hall or road.
- **Avoid-static-obstacle**: move away from non-threatening obstacles.
- **Dodge**: sidestep approaching ballistic objects.
- **Escape**: Evade intelligent predators.
- **Noise**: move in a random direction for a fixed amount of time. (persistence)
- **Avoid-past**: move away from recently visited areas.
- **Probe**: move towards an open area.
- **Dock**: move in a spiral trajectory towards a particular surface.
- **Teleautonomy** - introduce a human operator at the same level as other behaviors.

Vector-fields for different behaviors (schemata)

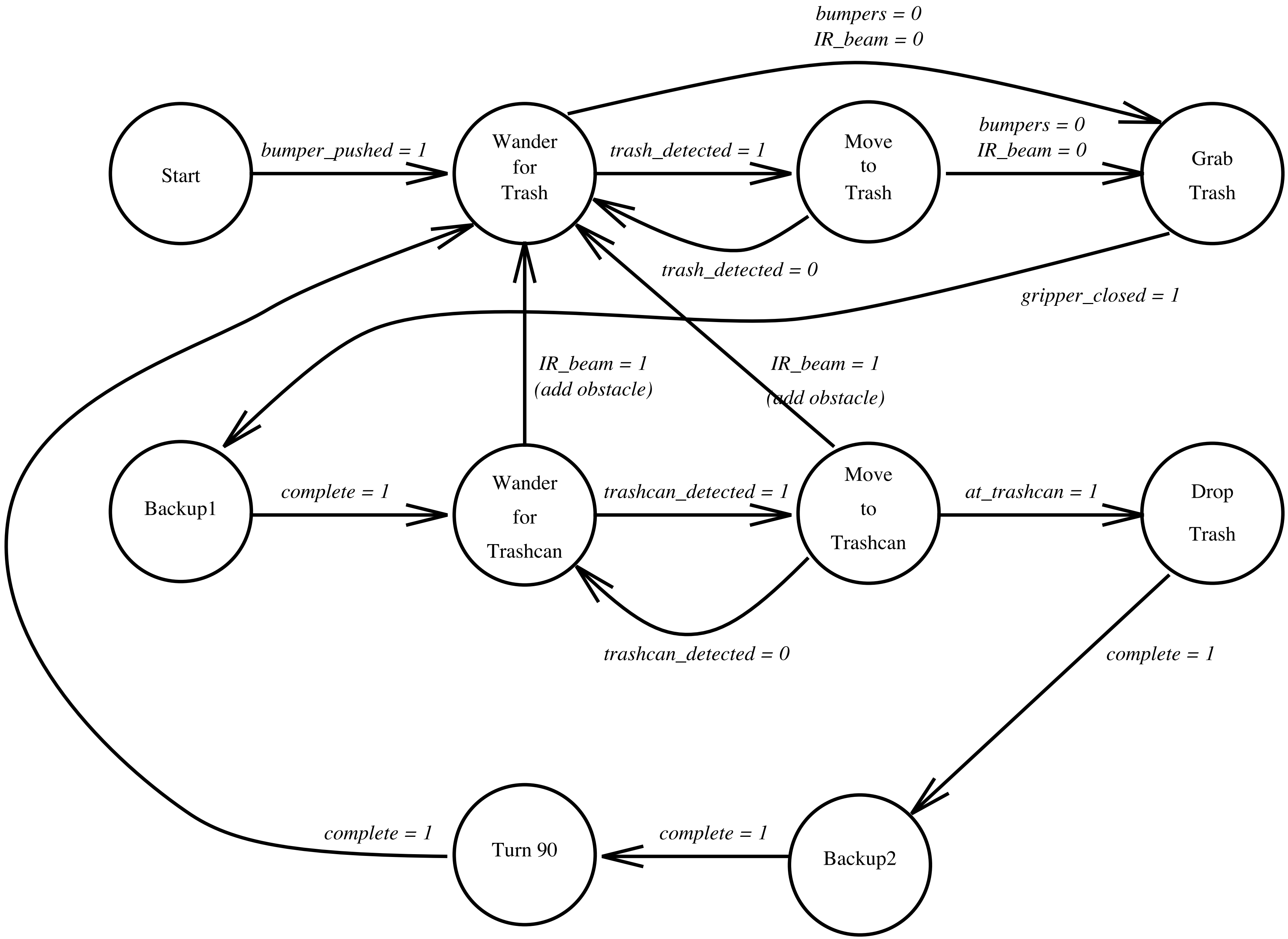


(A) (B) (C) (D)

Superposing potential fields to combine behaviors

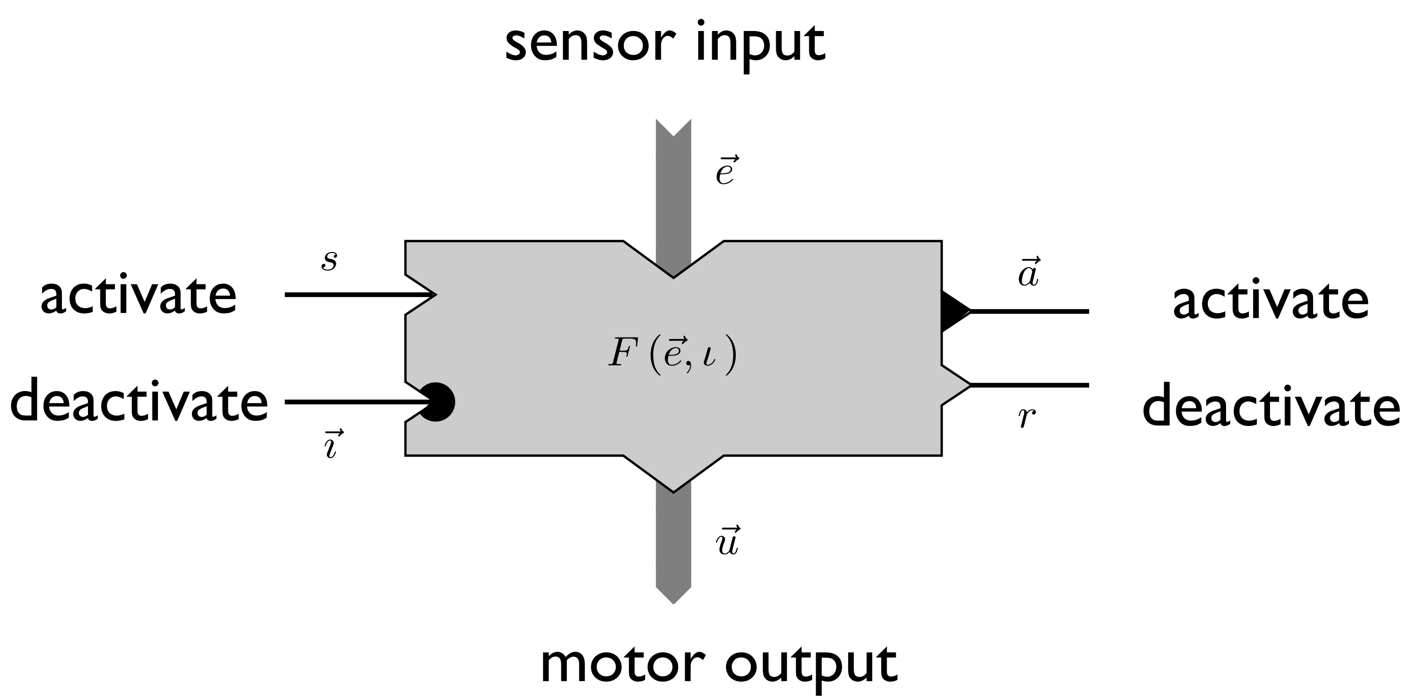


Behavior-based sequence planner



Scaling behavior-based architectures

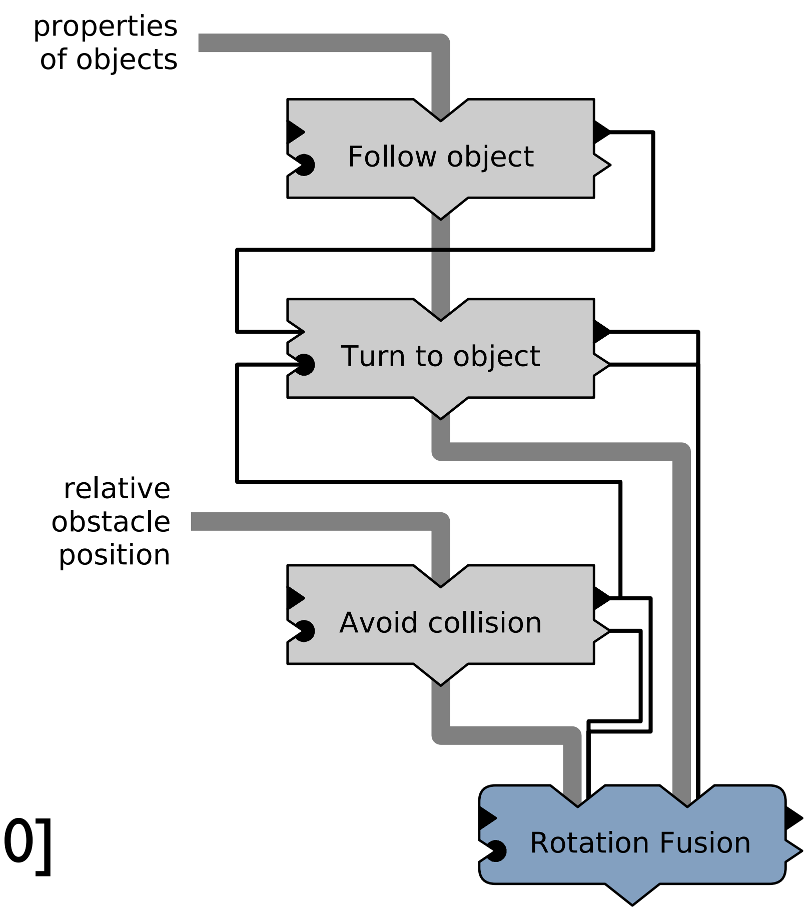
- behaviors: map sensor input to motor output
- are activated/deactivated
- and may in term activate/deactivate other behaviors



[Proetzsch, Luksch, Berns 2010]

Scaling behavior-based architectures

- behaviors: map sensor input to motor output
- are activated/deactivated
- and may in term activate/deactivate other behaviors



[Proetzsch, Luksch, Berns 2010]

Scaling behavior-based architectures

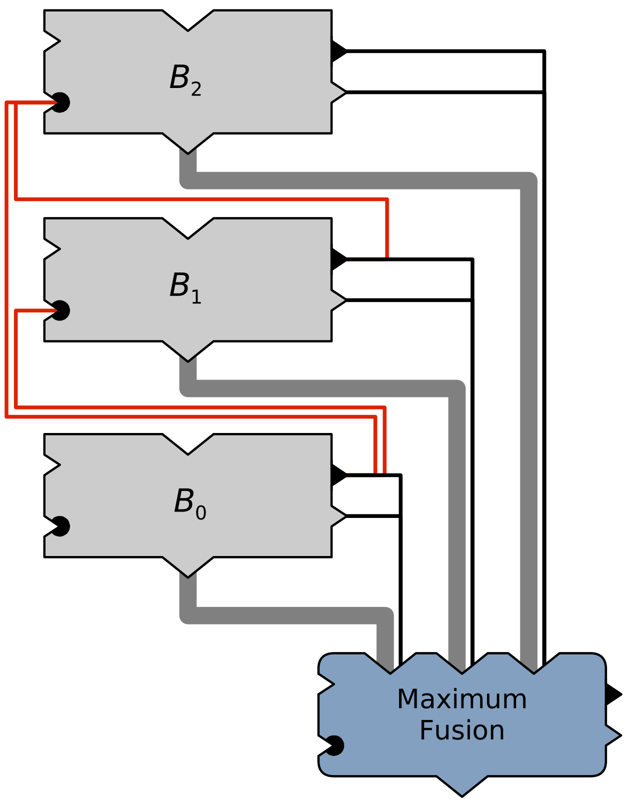


Fig. 5. Priority-based arbitration in iB2C.

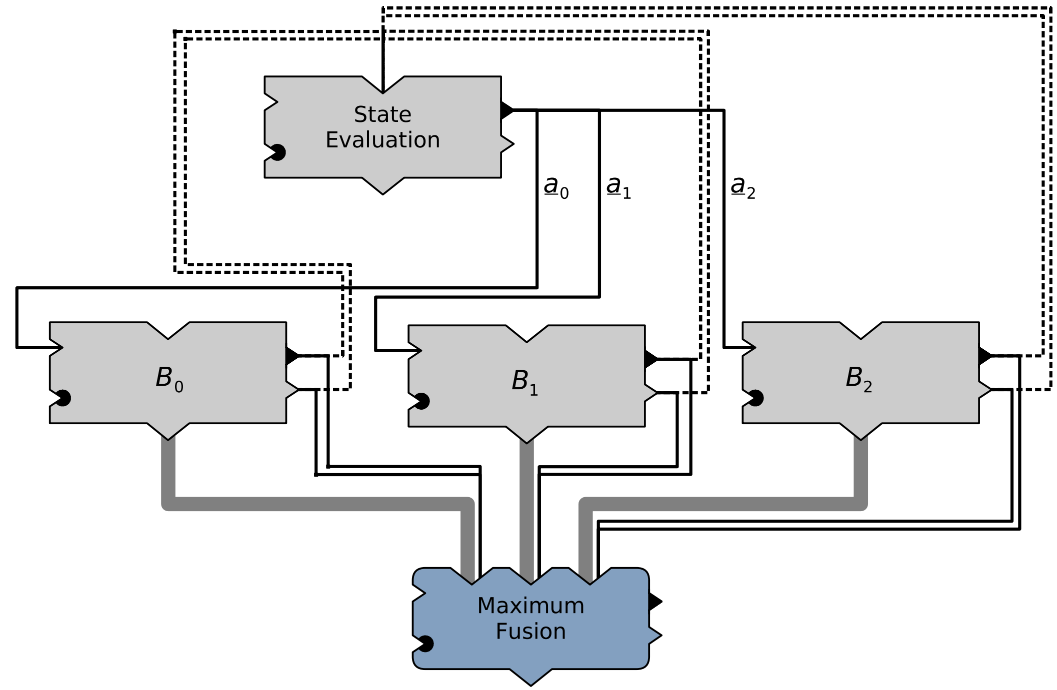


Fig. 6. State-based arbitration in iB2C.

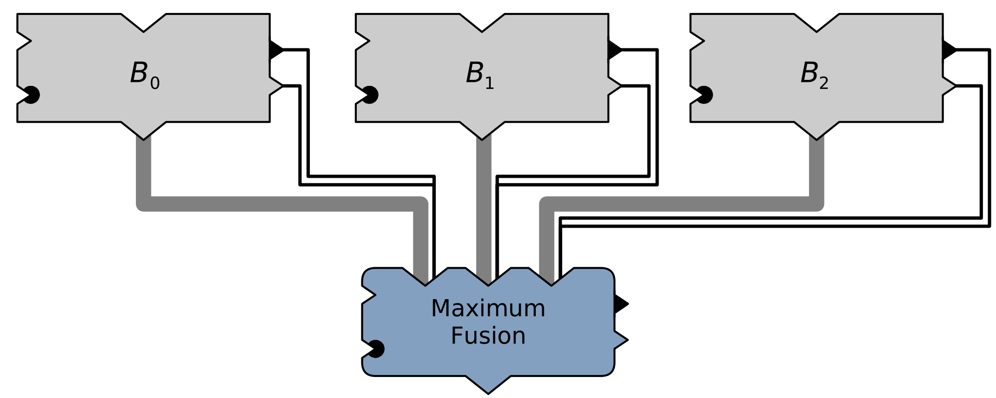
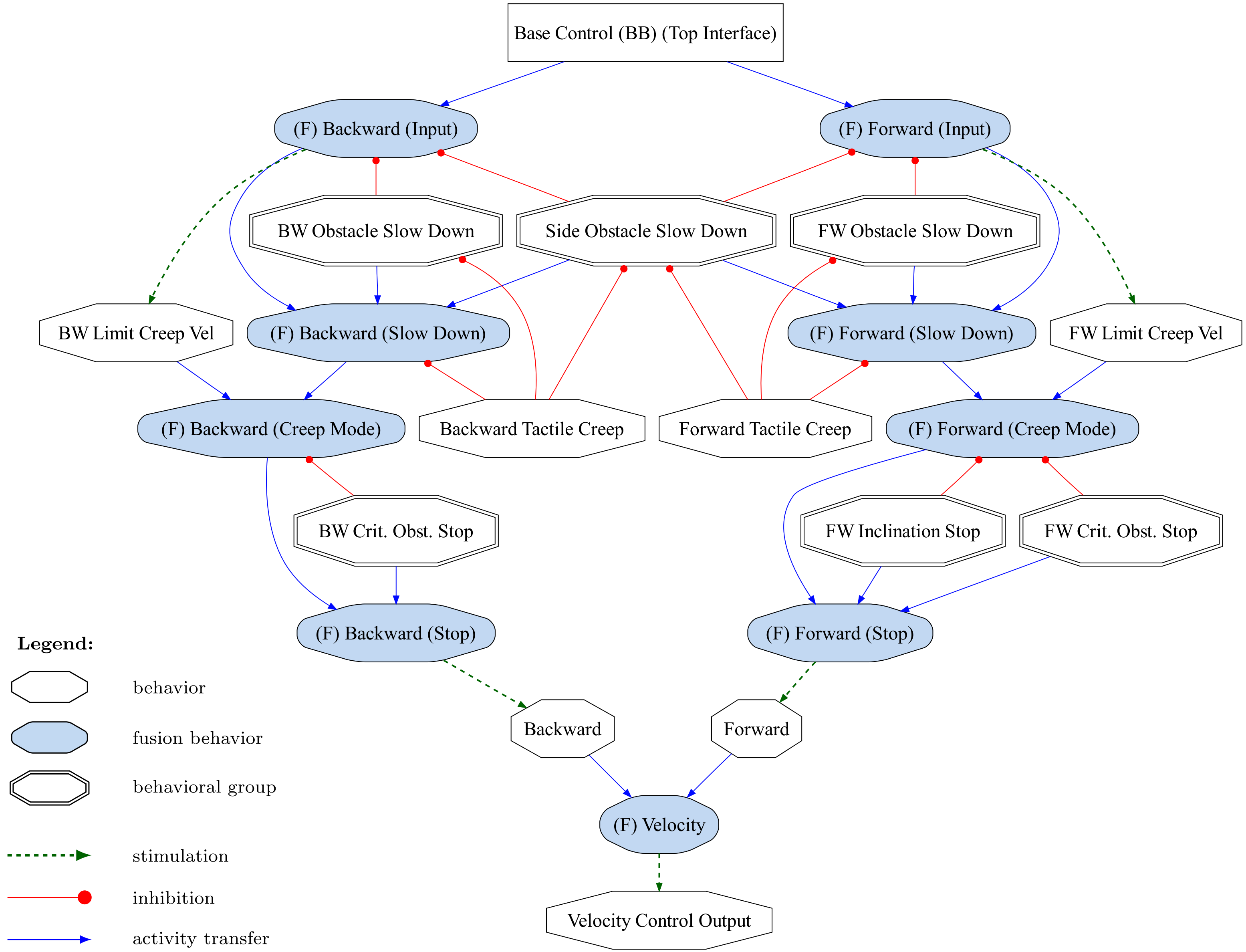


Fig. 7. Winner-takes-all arbitration in iB2C.

[Proetzsch, Luksch, Berns 2010]

Scaling behavior-based architectures



[Proetzsch, Luksch, Berns 2010]

Scaling behavior-based architectures

- implemented on a variety of systems

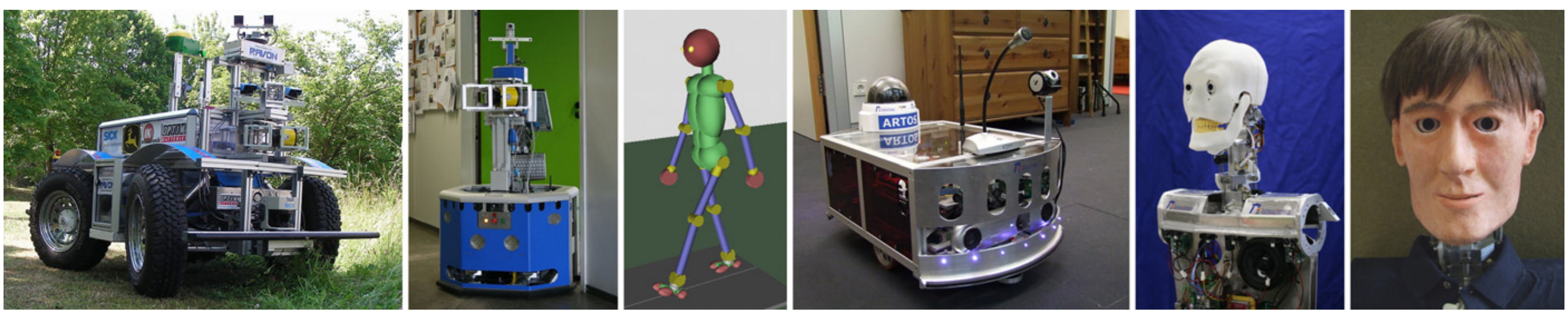
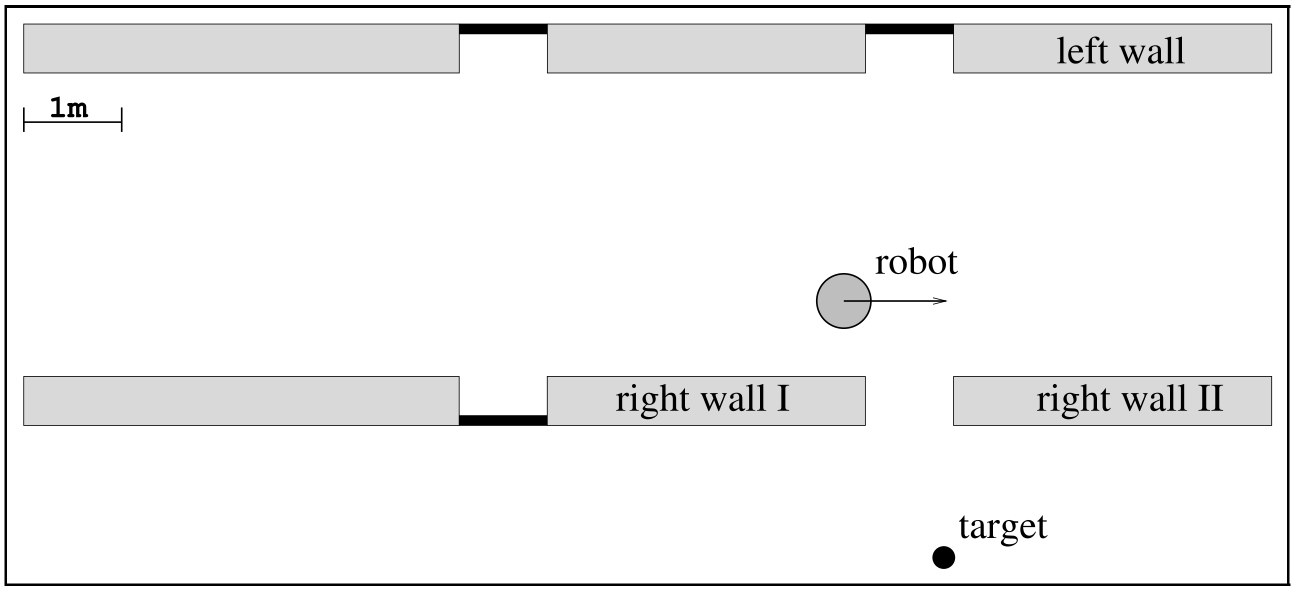


Fig. 20. Robots of the Robotics Research Lab controlled by an iB2C system: Ravon, Marvin, dynamically simulated biped, Artos, and Roman (skeleton and skin).

[Proetzsch, Luksch, Berns 2010]

Dynamic window approach

- take dynamic constraints of vehicle into account (maximal decelerations/accelerations)… to drive fast



[Fox, Burghard, Thrun, 1996]

Dynamic window approach

- discretize motor control space: linear and angular velocity
- => search space: circular trajectories of v, omega

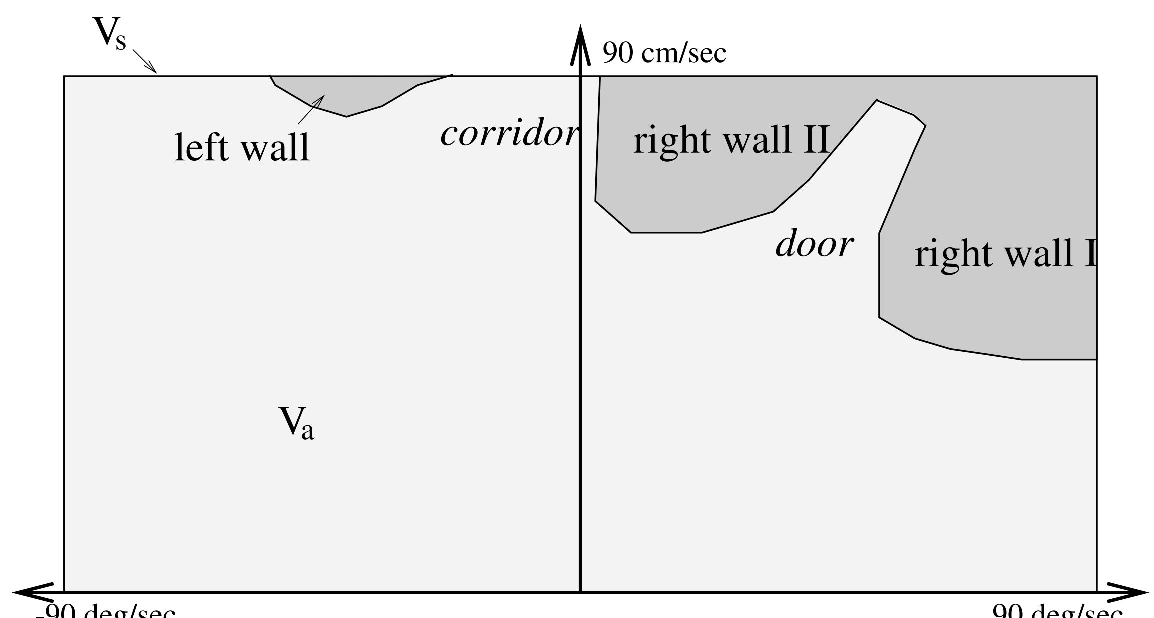


Figure 4. Velocity space

Dynamic window approach

1. **Search space:** The search space of the possible velocities is reduced in three steps:
   (a) **Circular trajectories:** The dynamic window approach considers only circular trajectories (curvatures) uniquely determined by pairs (v, ω) of translational and rotational velocities. This results in a two-dimensional velocity search space.
   (b) **Admissible velocities:** The restriction to admissible velocities ensures that only safe trajectories are considered. A pair (v, ω) is considered admissible, if the robot is able to stop before it reaches the closest obstacle on the corresponding curvature.
   (c) **Dynamic window:** The dynamic window restricts the admissible velocities to those that can be reached within a short time interval given the limited accelerations of the robot.

Dynamic window approach

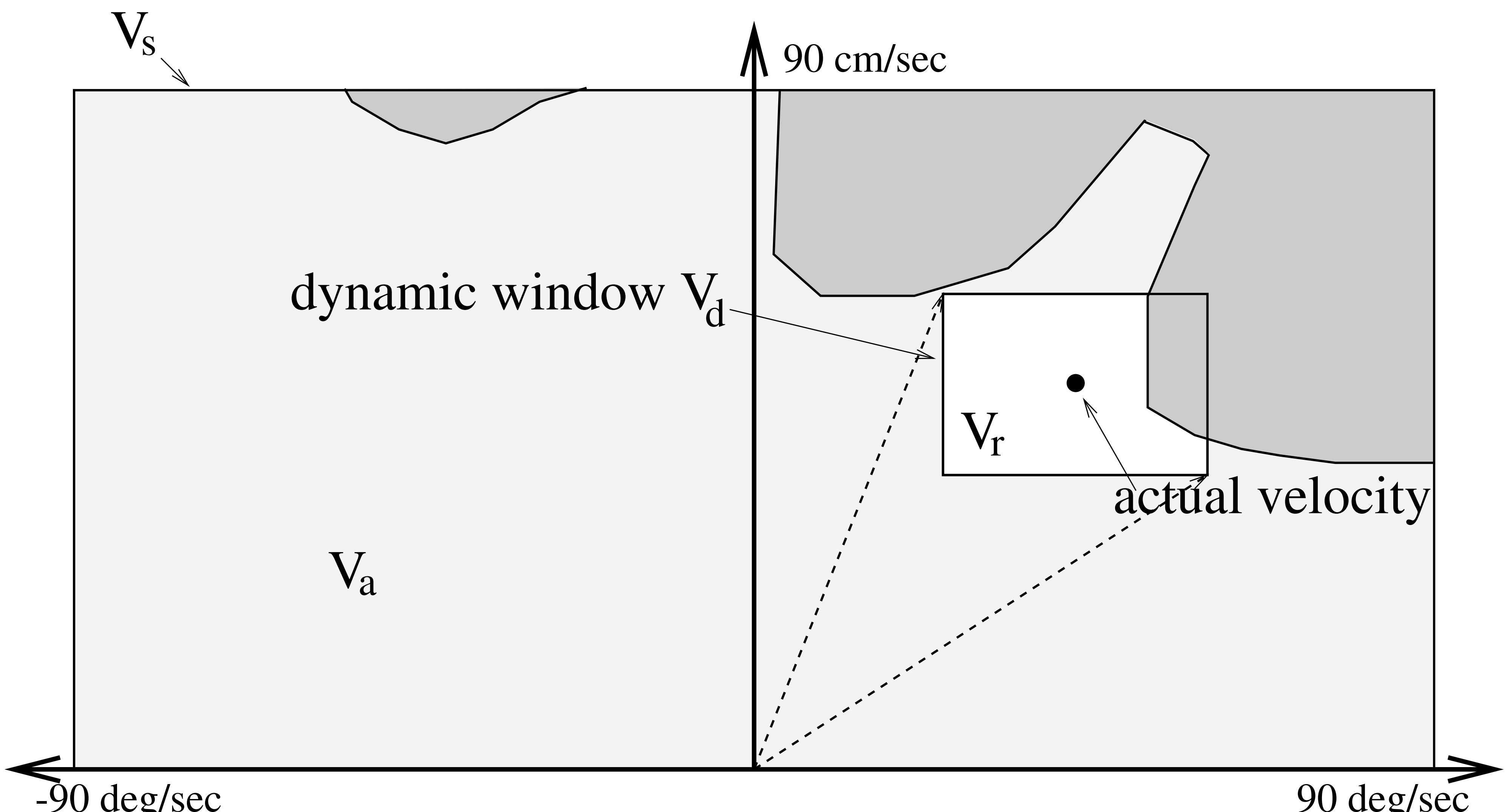


Figure 5. Dynamic window

Dynamic window approach

2. **Optimization:** The objective function

$$G(v, \omega) = \sigma(\alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{vel}(v, \omega)) \qquad (13)$$

is maximized. With respect to the current position and orientation of the robot this function trades off the following aspects:

   (a) **Target heading:** *heading* is a measure of progress towards the goal location. It is maximal if the robot moves directly towards the target.

   (b) **Clearance:** *dist* is the distance to the closest obstacle on the trajectory. The smaller the distance to an obstacle the higher is the robot's desire to move around it.

   (c) **Velocity:** *vel* is the forward velocity of the robot and supports fast movements.

The function σ smoothes the weighted sum of the three components and results in more side-clearance from obstacles.

Dynamic window approach

- target cost function

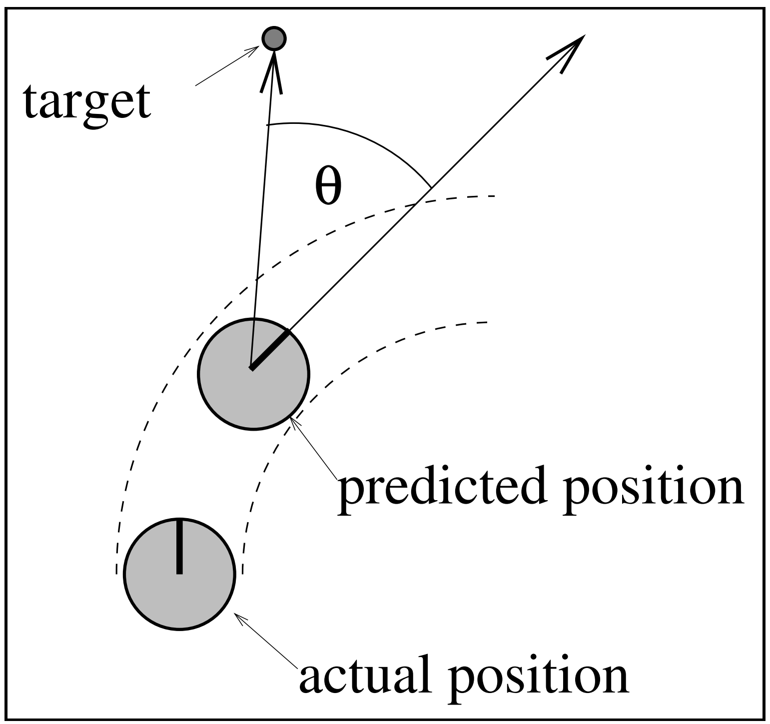


Figure 6. Angle θ to the target

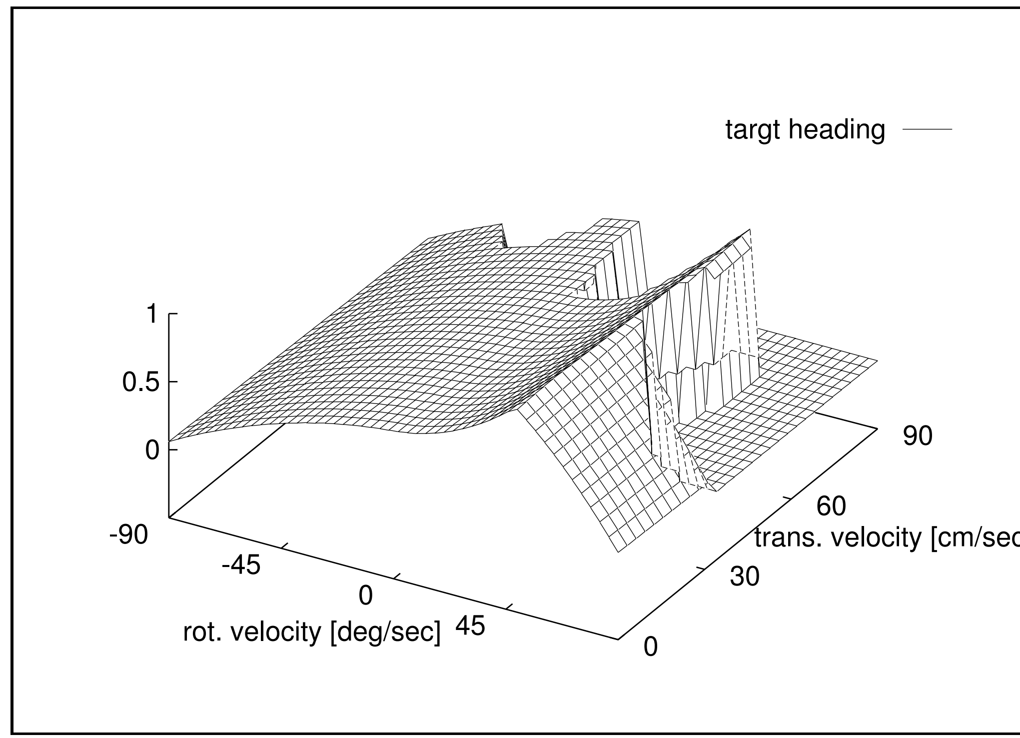


Figure 7. Evaluation of the target heading

Dynamic window approach

- clearance cost function

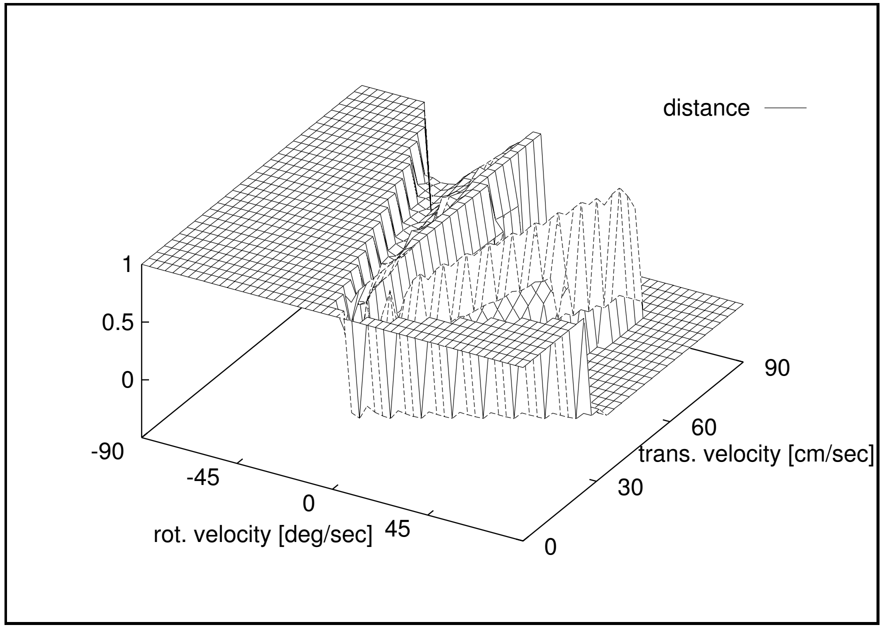


Figure 8. Evaluation of the distances

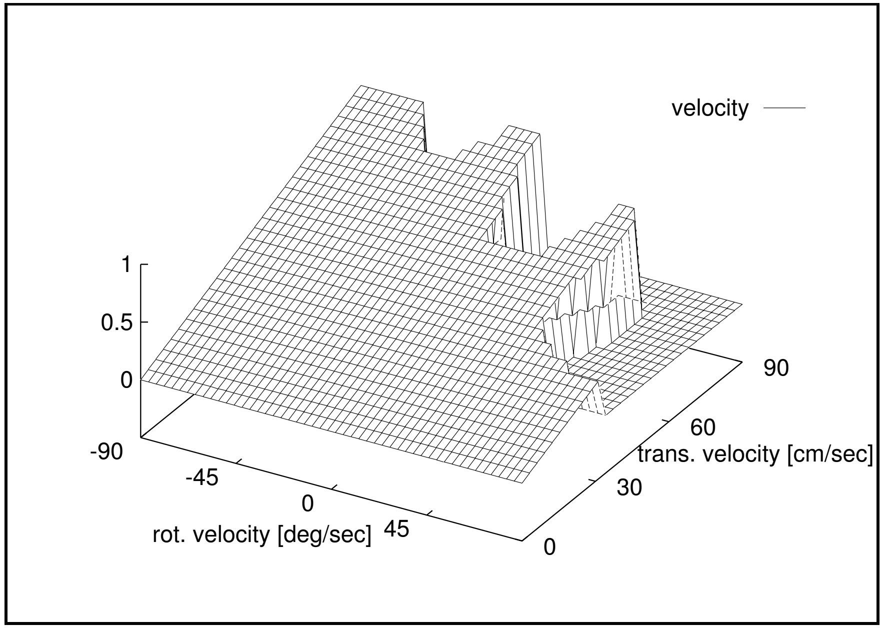


Figure 9. Evaluation of the velocities

Dynamic window approach

- smoothing the cost functions

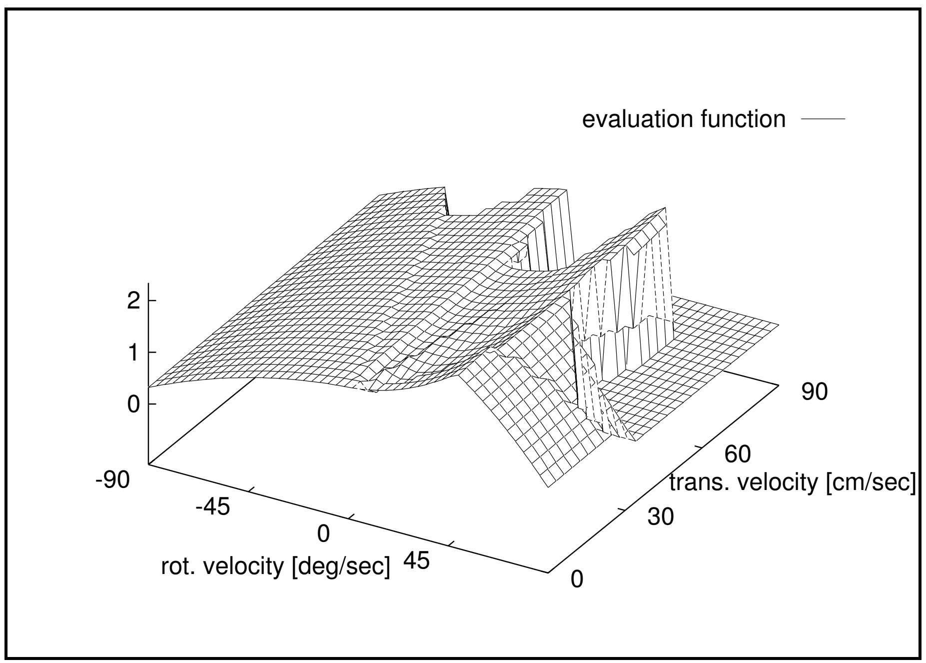


Figure 10. Combined evaluation function

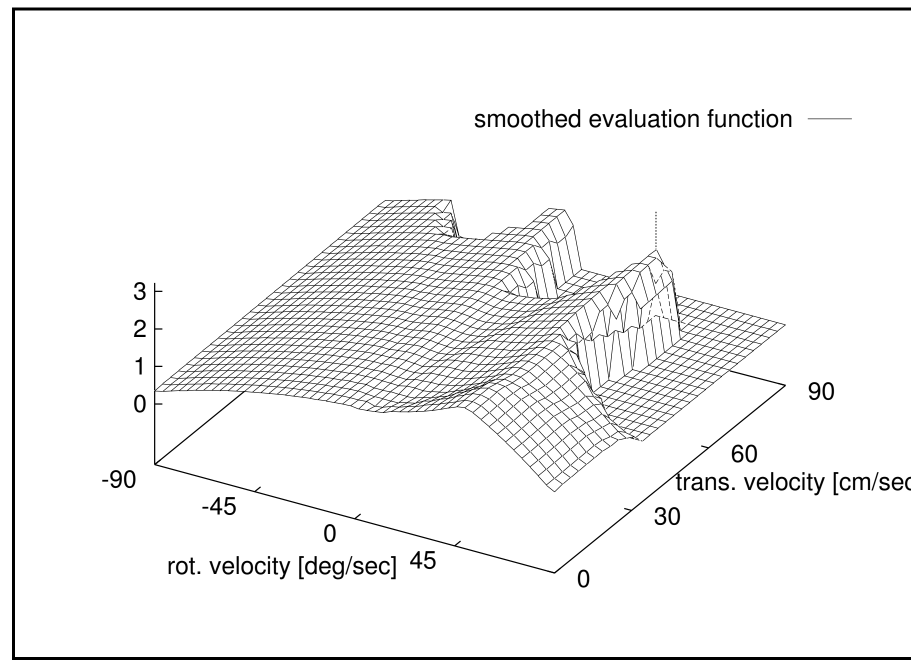


Figure 11. Objective function

Dynamic window approach

- two samples of actual velocities

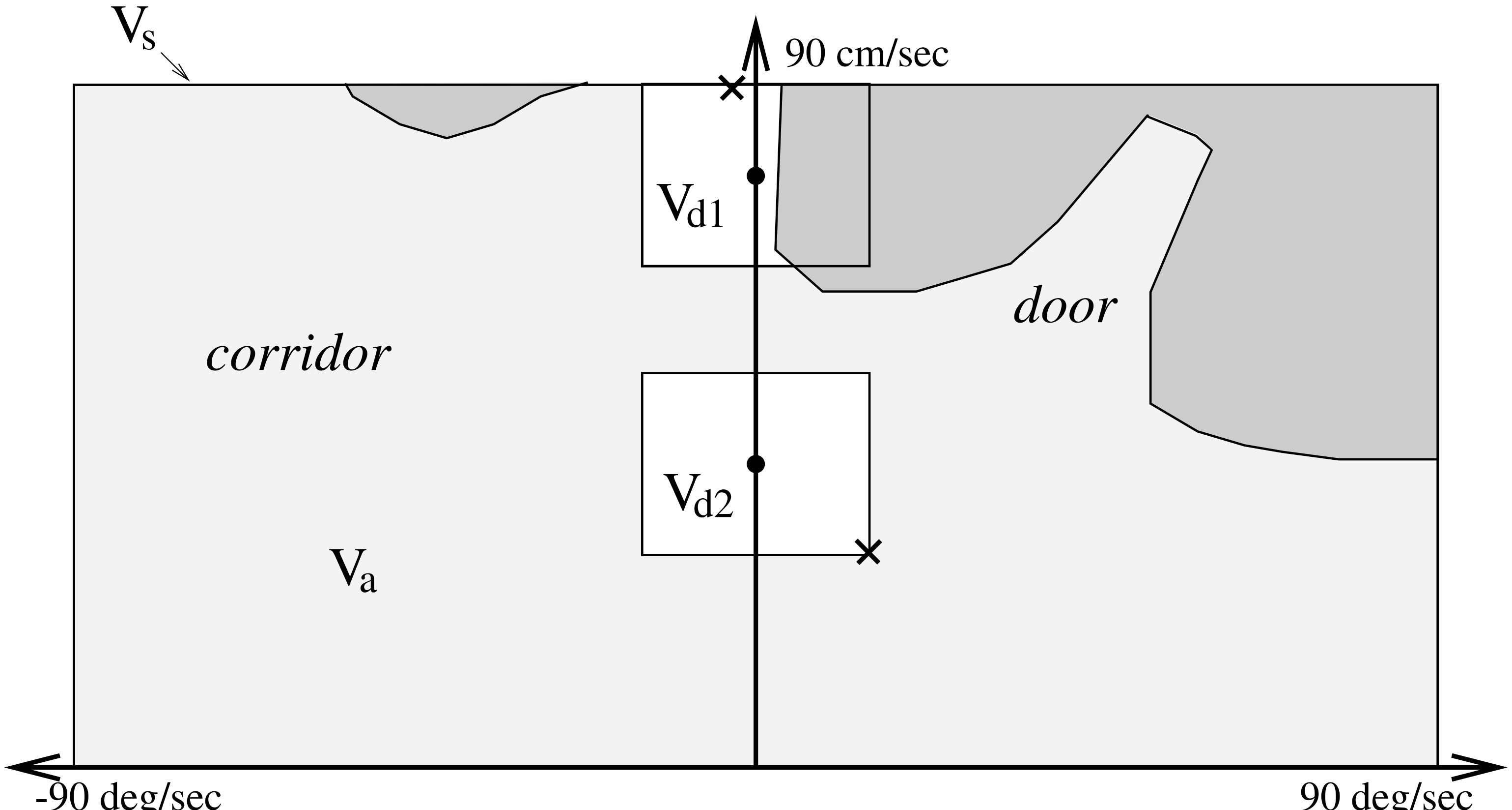


Figure 12. Velocity space

Dynamic window approach

- cost function for the action velocities

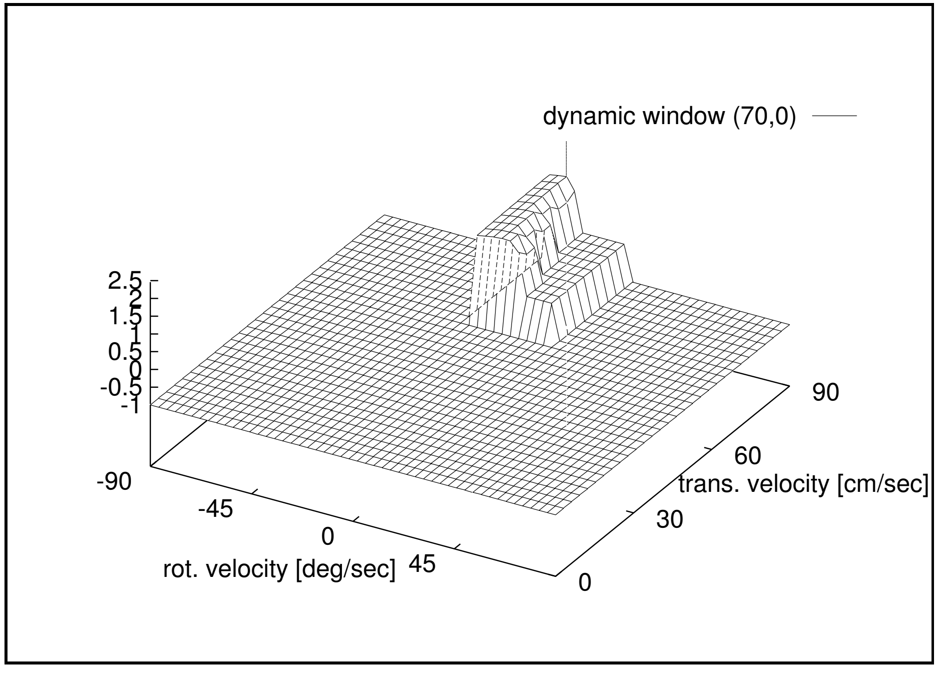


Figure 13. Objective function for actual velocity (75,0)

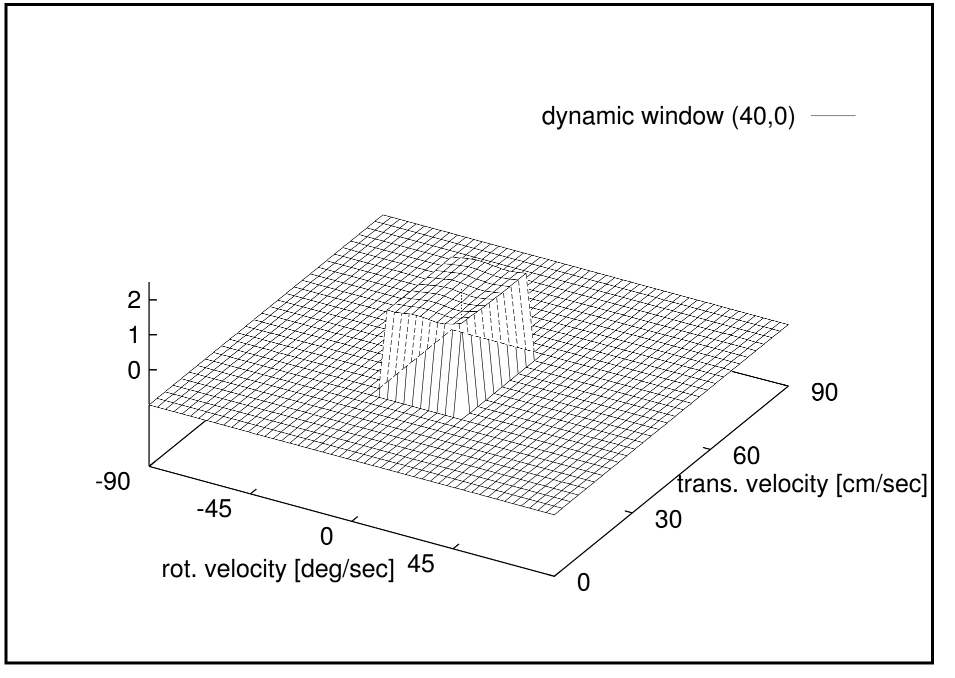


Figure 14. Objective function for actual velocity (40,0)

Dynamic window approach

- example RHINO
- used Borenstein Koren approach to smooth and accumulate sonar distance data

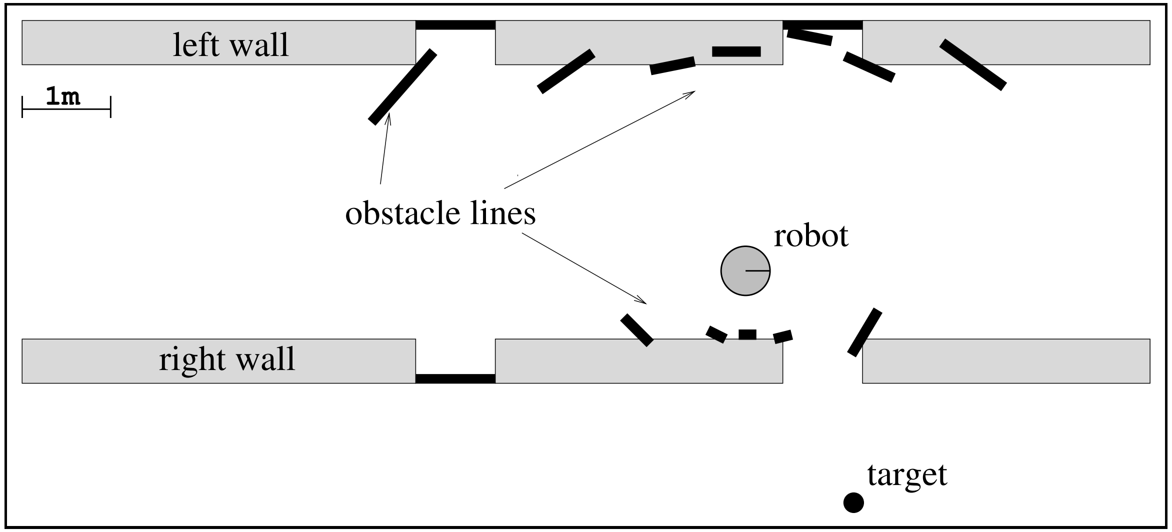


Figure 18. Example environment with obstacle lines and target point

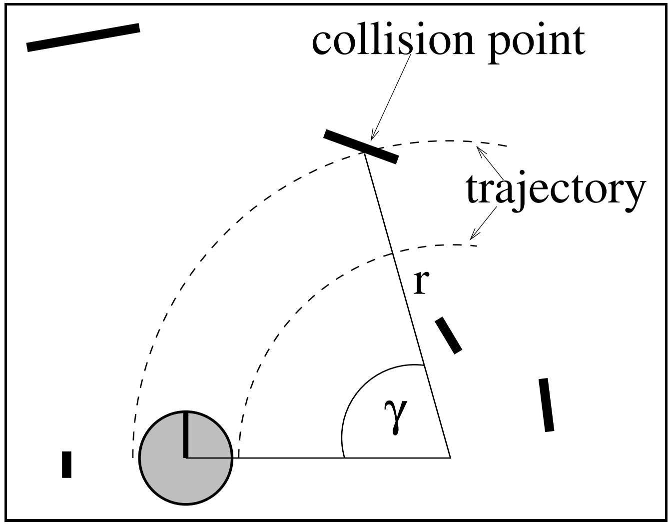


Figure 19. Determination of the distance

Dynamic window approach

- data

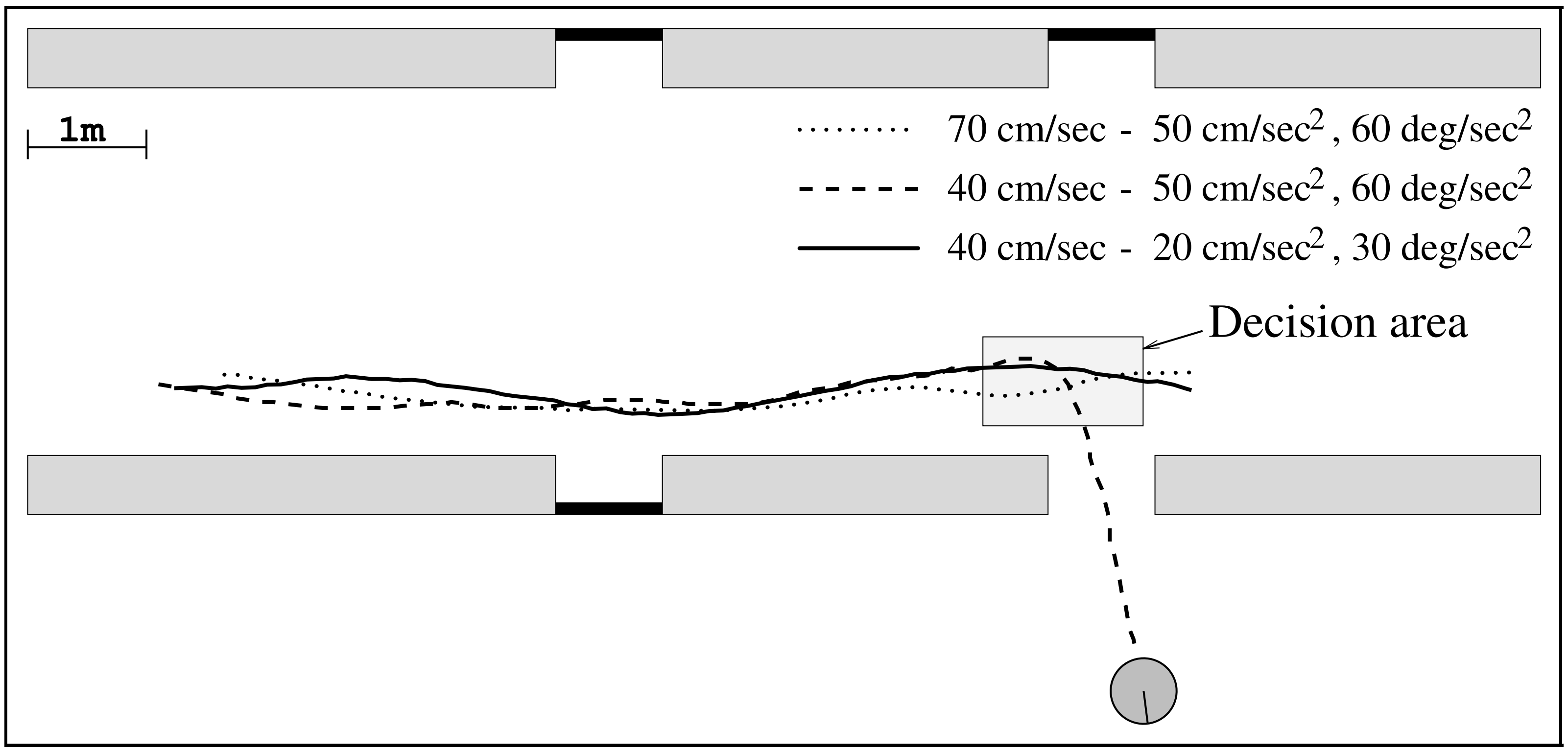


Figure 20. Trajectories chosen for different dynamic parameters

Dynamic window approach

- data

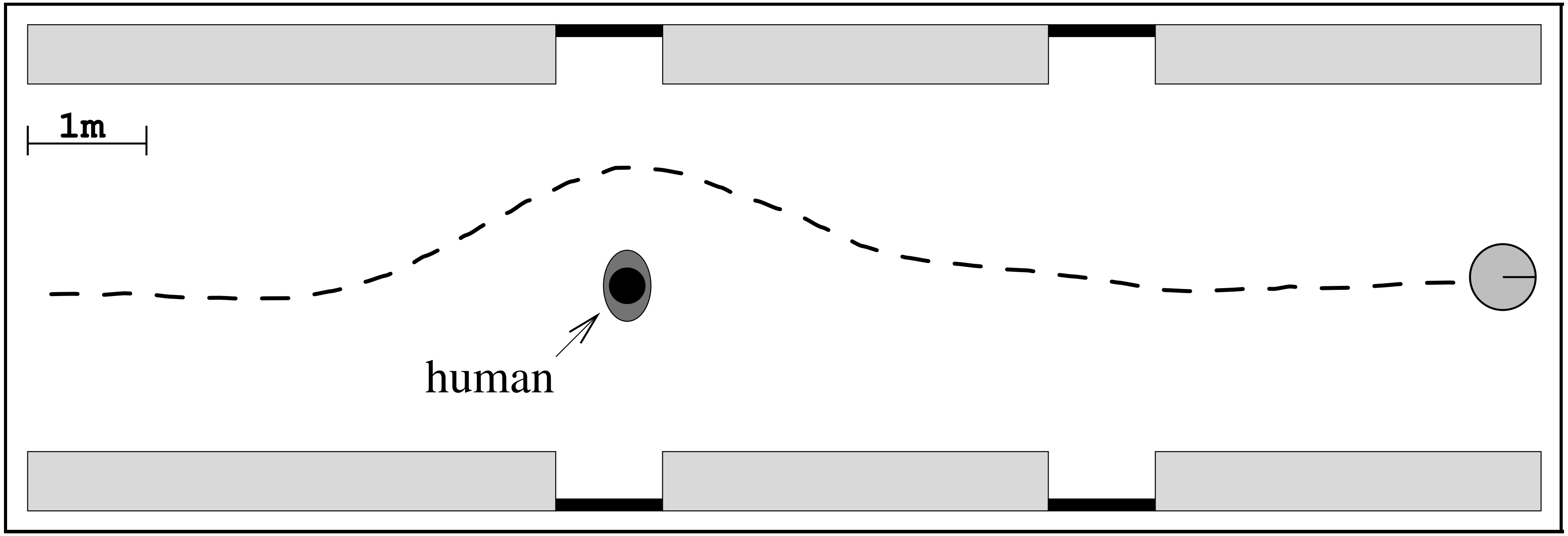


Figure 21. Trajectory through corridor

Dynamic window approach

- data

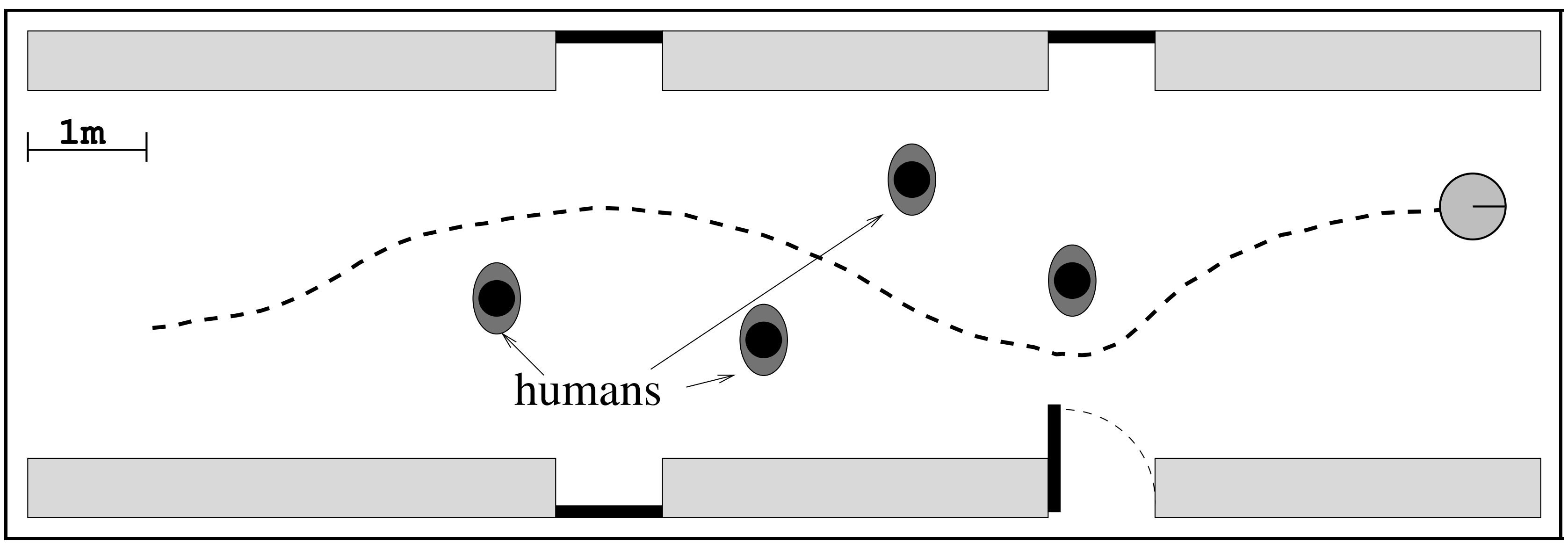


Figure 22. Trajectory through cluttered corridor

Summary

- powerful approaches exist for motion planning
- the best/exact approaches make strong demands on world representations and computation
- heuristic "reactive" approaches are state of the art (often combined in hybrid architectures with deliberative planning)
- the attractor dynamics approach is competitive as a reactive approach

Outlook

- deliberative planning…
  - moving beyond the vehicle navigation problem
  - planning sequences of actions to achieve goals
  - searching spaces, often represented as graphs
  - … a huge field…
- not very satisfactorily included in neurally based approaches..