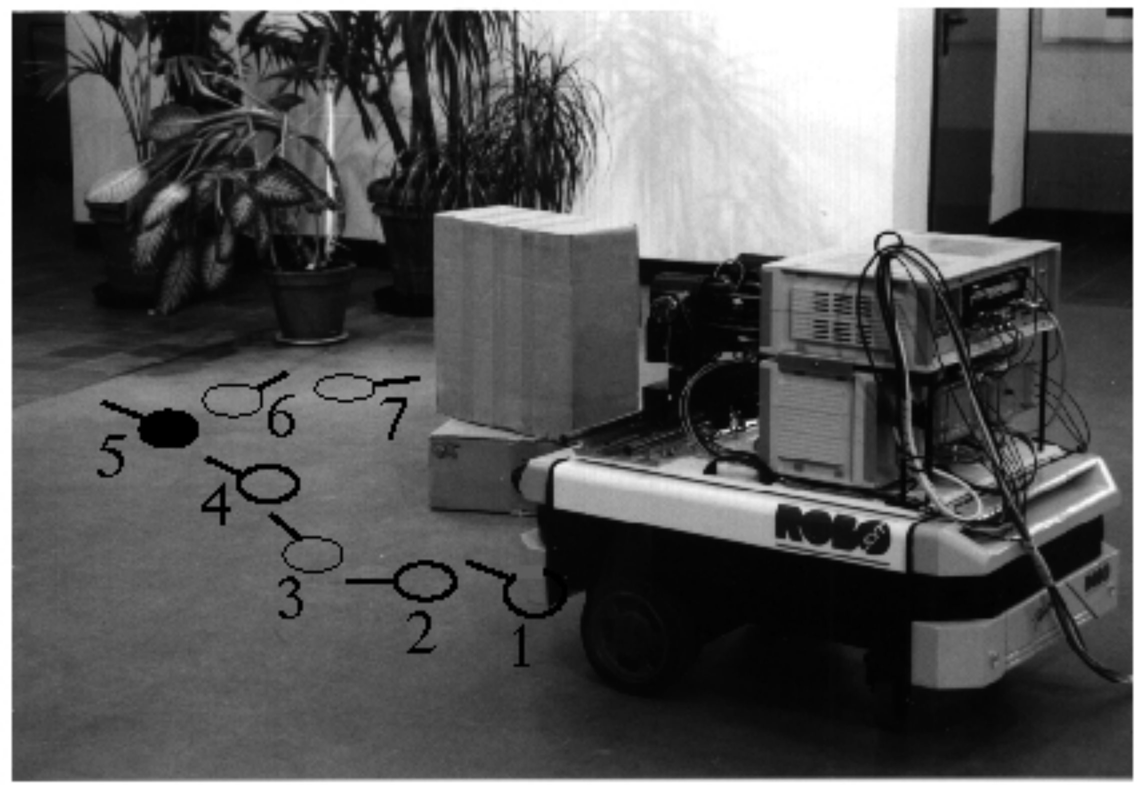
Vehicle motion planning: Survey over approaches

Gregor Schöner
INI RUB Germany

The problem

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



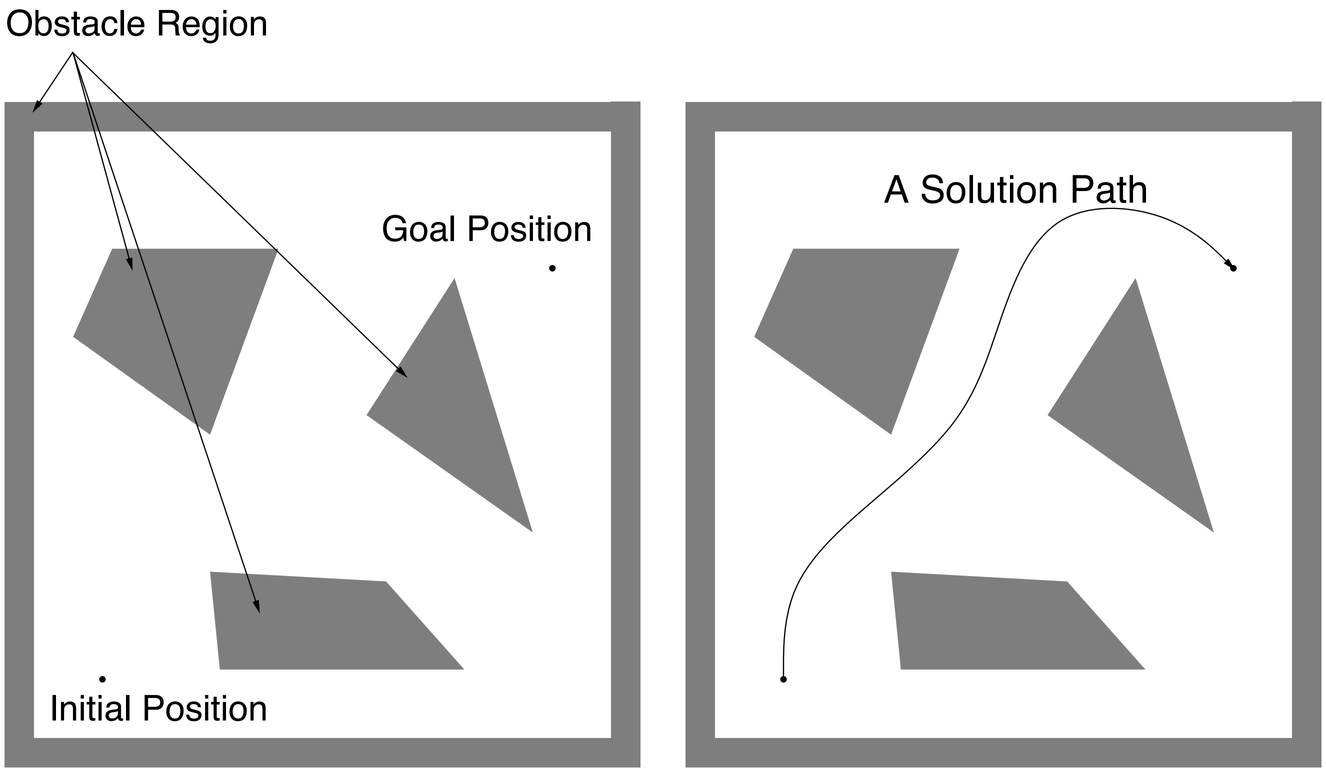
Approaches to vehicle path planning

- classical planning approaches
- potential field approach
- compared to attractor dynamics approach
- Borenstein & Koren
- extending attractor dynamics approach
- Dynamic window approach

Classical path planning

References

- Latombe: Robot motion planning, 1991
- LaValle: Planning algorithms, 2006, 2010
- Kavraki, LaValle: Chapter 7 of Springer Handbook of Robotics 2016

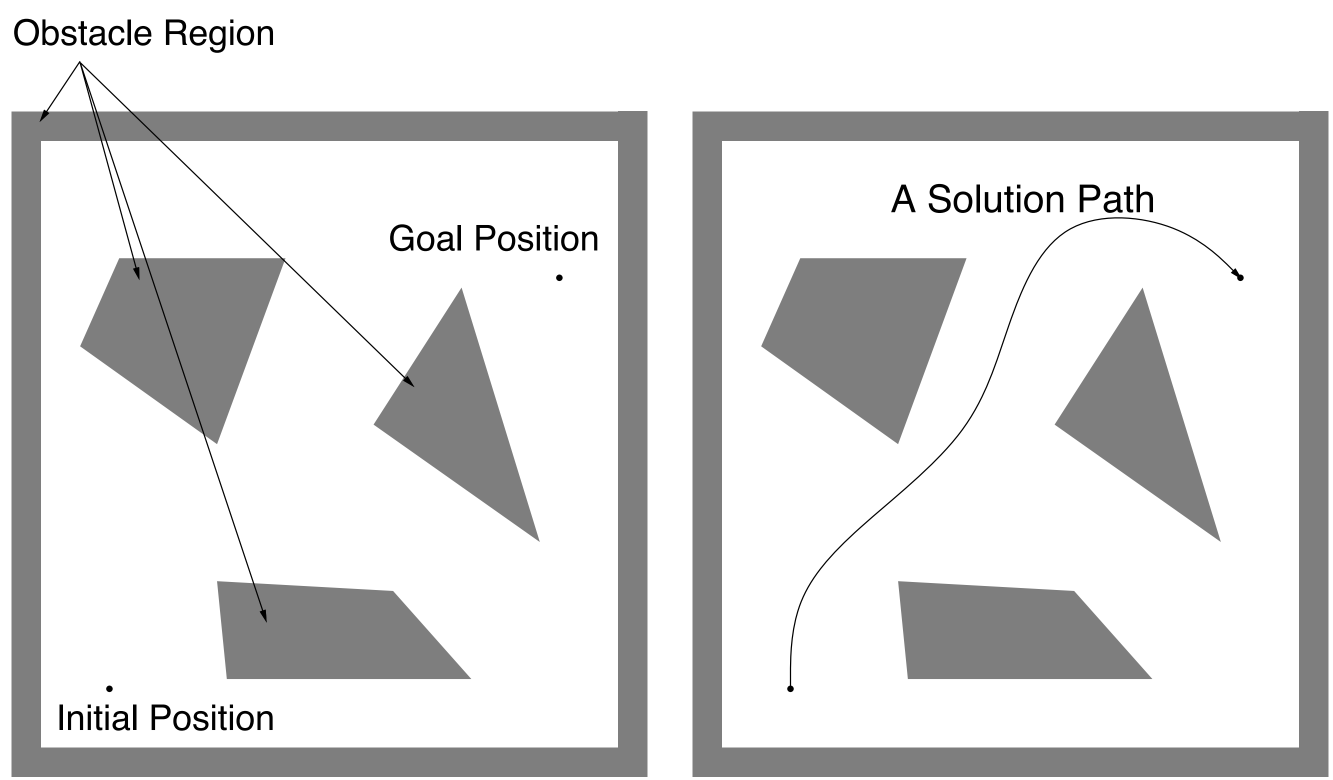


(a) (b)

[LaValle, 2006]

Classical motion planning

- based on a model of the environment (obstacles) and the robot

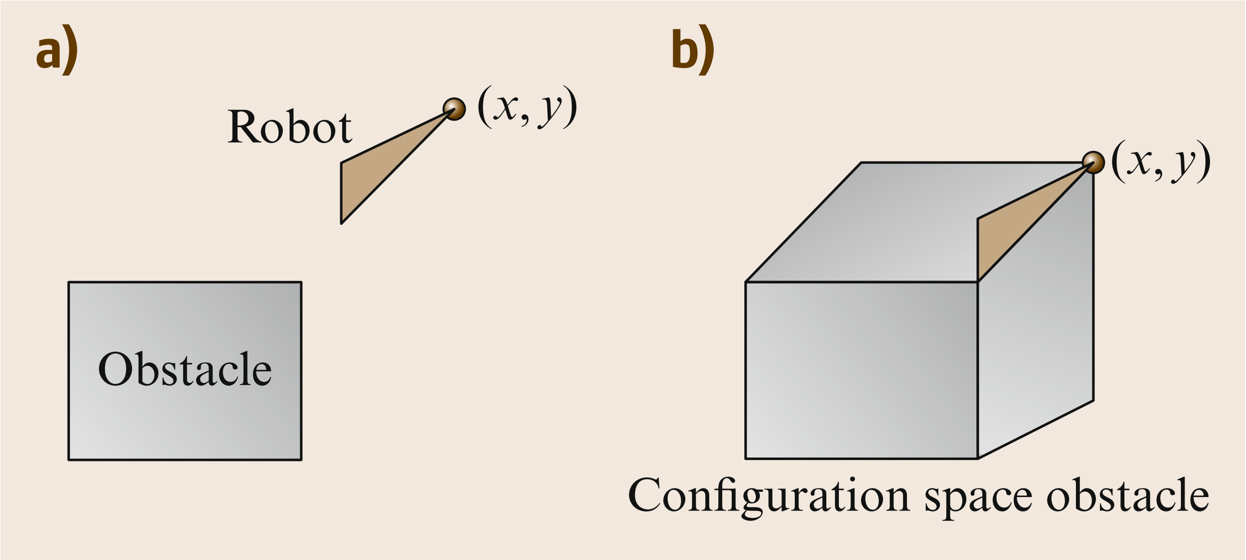


(a) (b)

Classical motion planning

- notion of configuration space

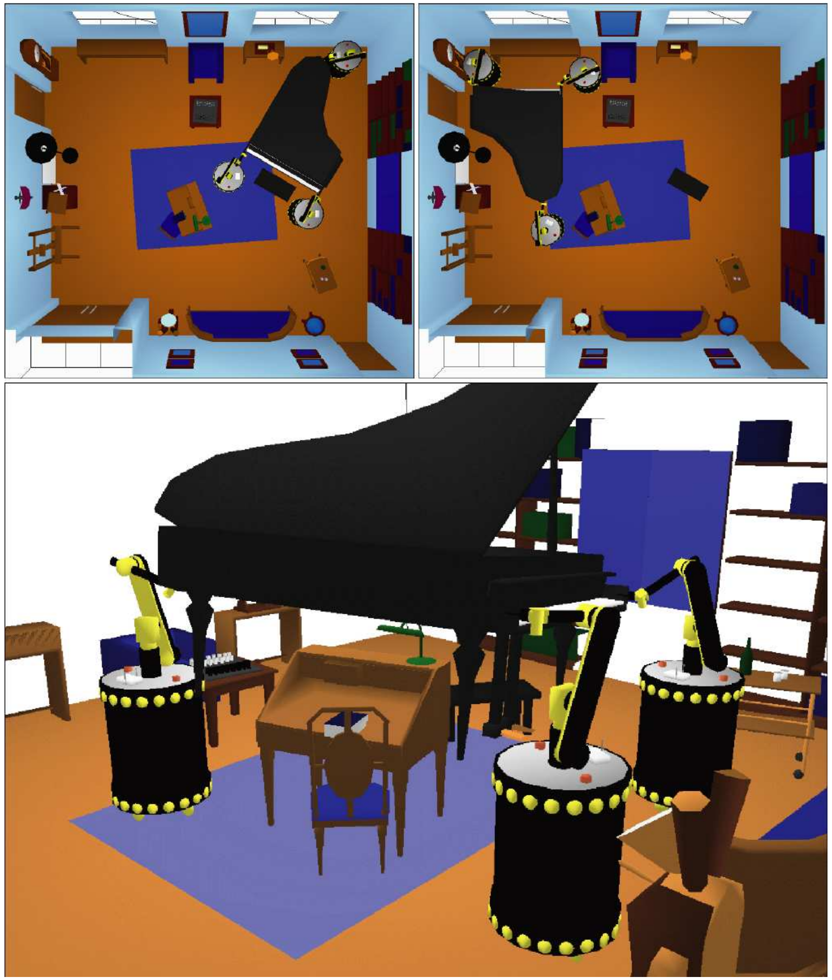
example:
square obstacle
triangular robot



[Kavraki, LaValle 2016]

Classical motion planning

- geometric path planning
  - “piano mover’s problem”
- computational complexity



[La Valle 2010]

Classical global path planning

- sampling based approaches
- road maps

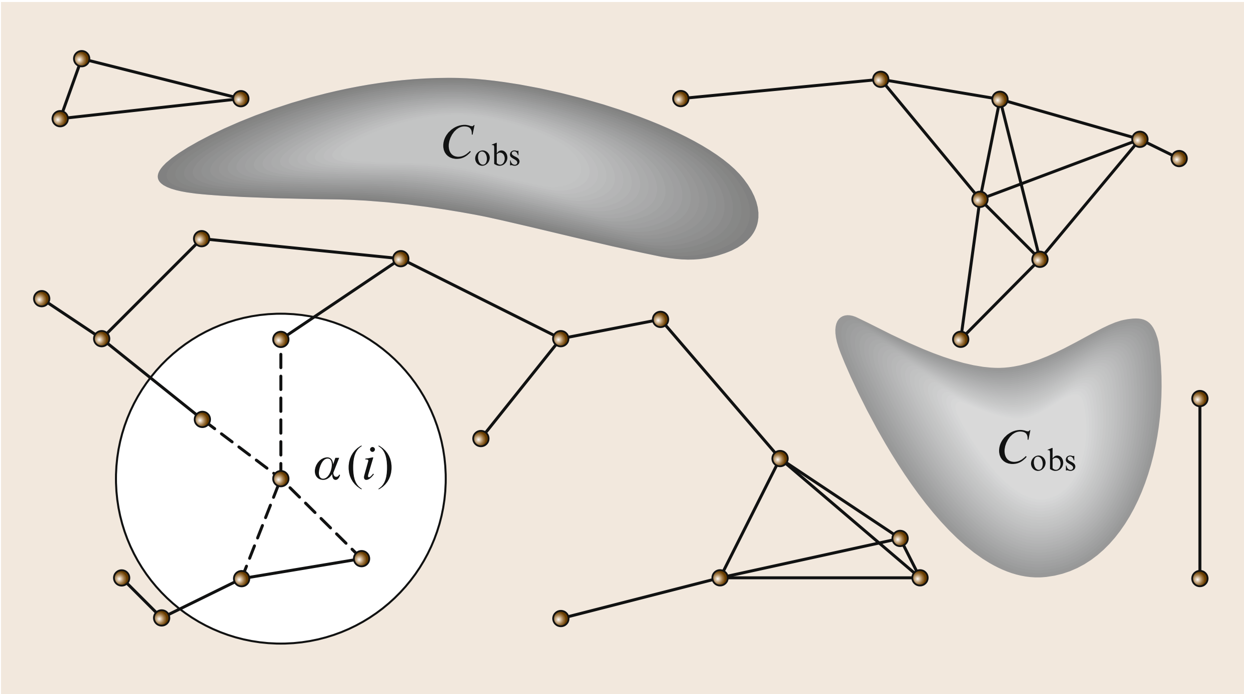


Fig. 7.3 The sampling-based roadmap is constructed incrementally by attempting to connect each new sample, $\alpha(i)$, to nearby vertices in the roadmap

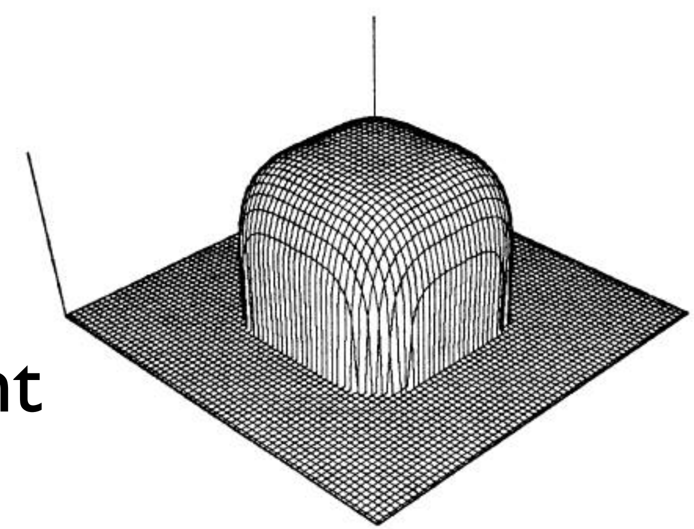
[Kavraki, LaValle 2016]

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



[Khatib, 1986]

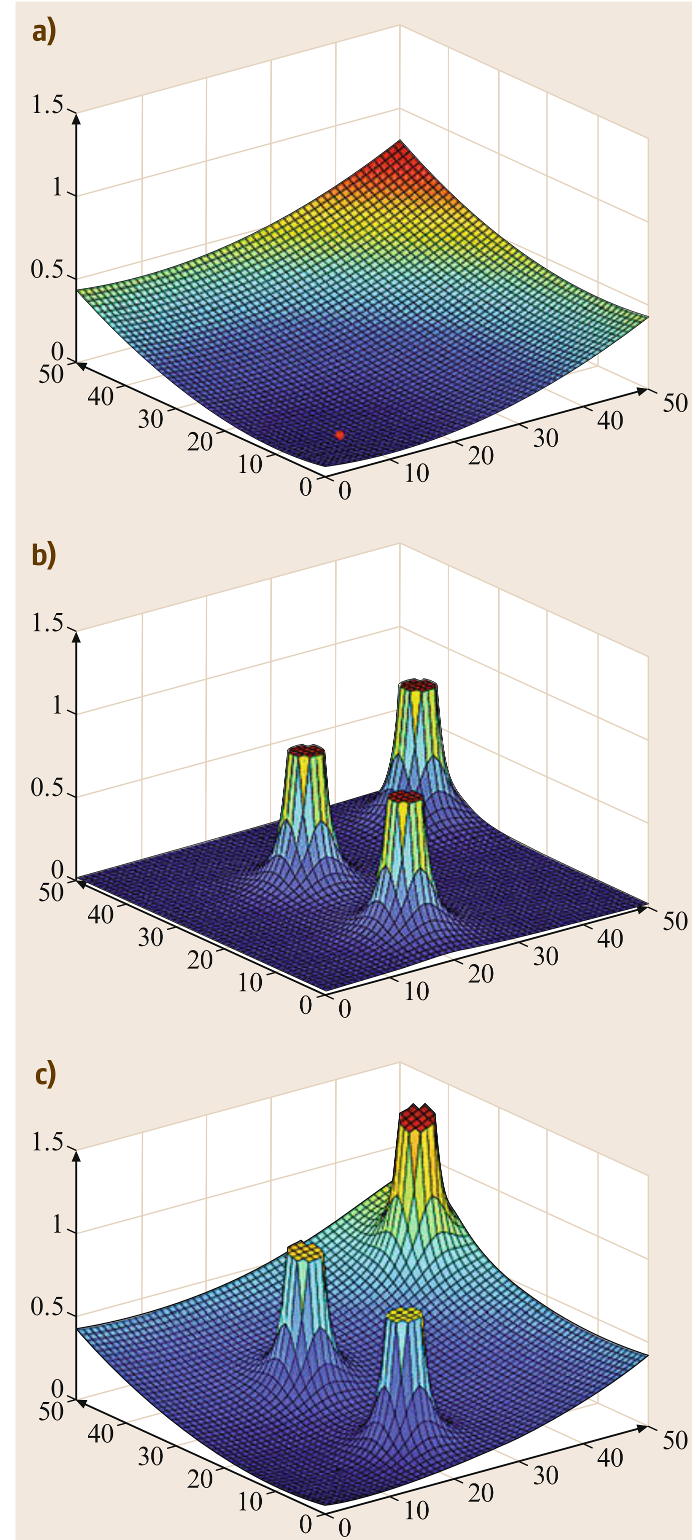
Potential field approach

target component

obstacle component

sum

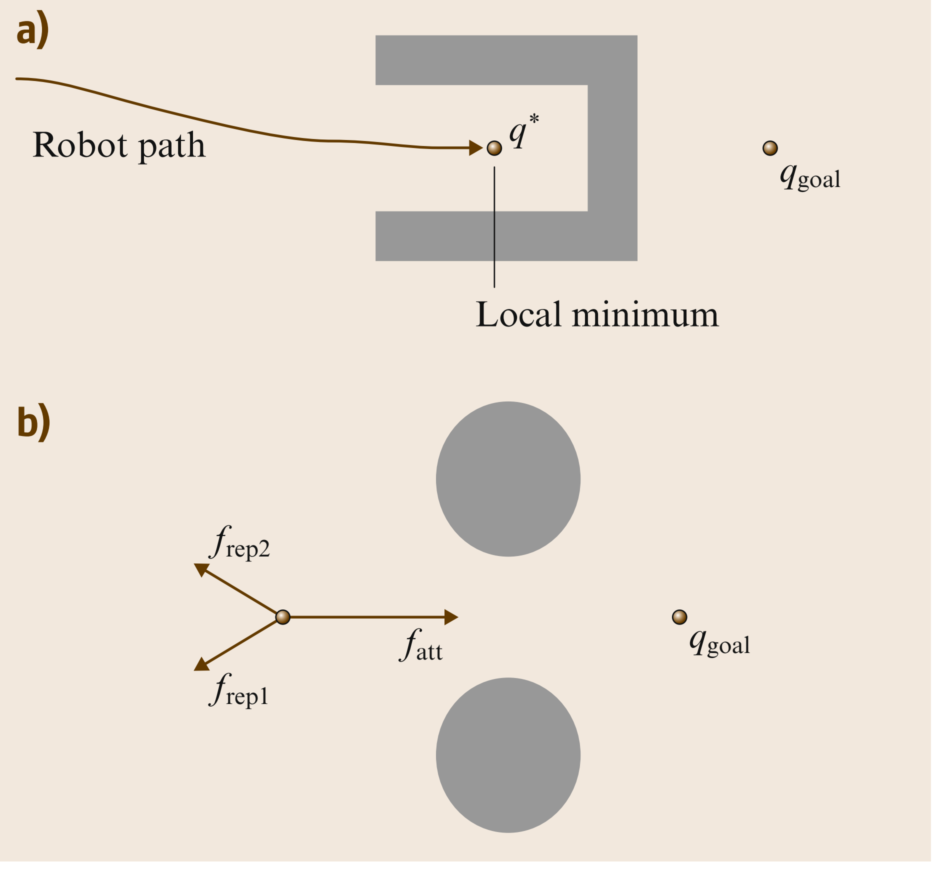
[Kavraki, LaValle 2016]



Potential field approach

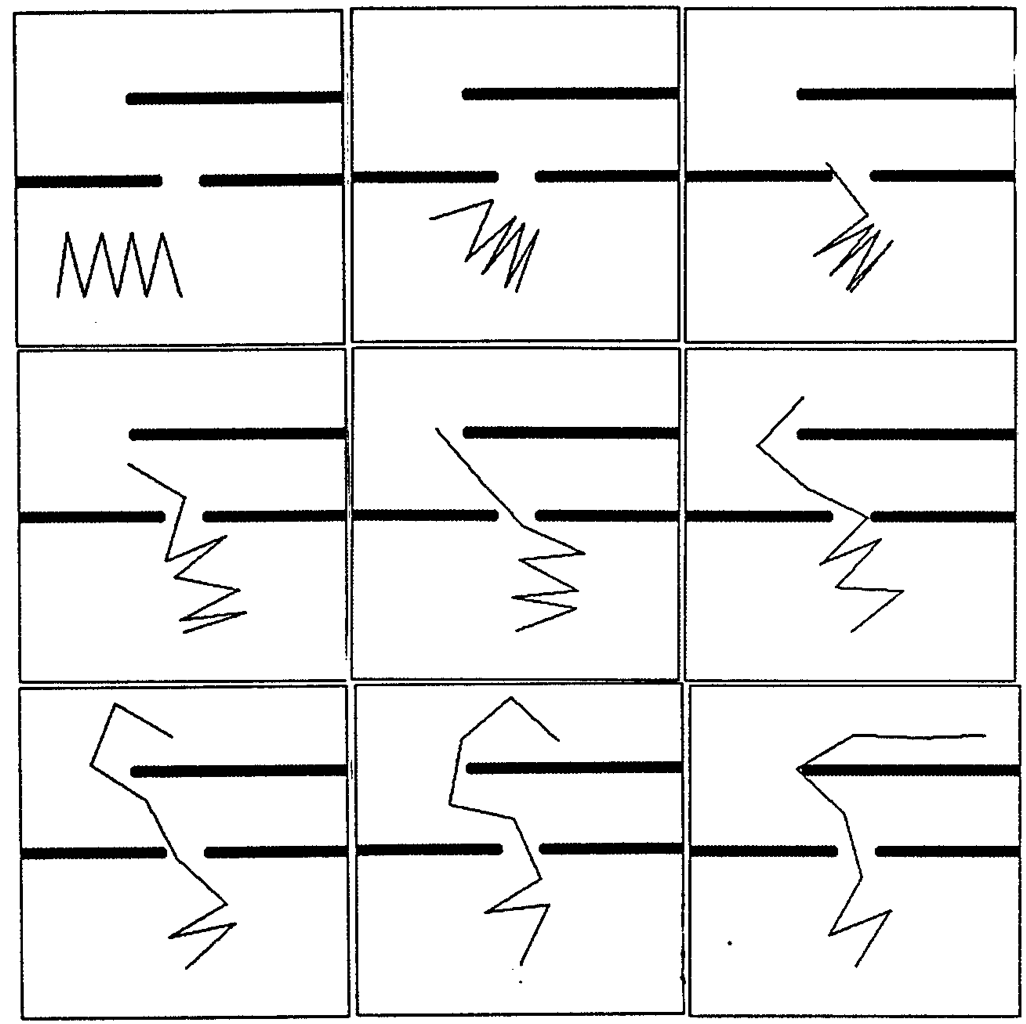
- heuristic approach: no guarantee
- problem of local minima

[Kavraki, LaValle 2016]



Potential field approach

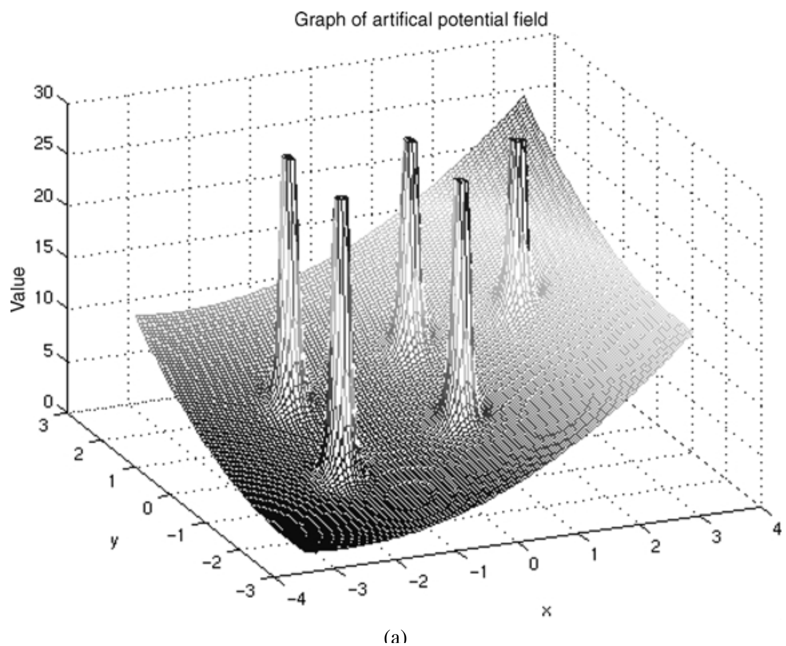
- generalization to higher-dimensional configuration spaces

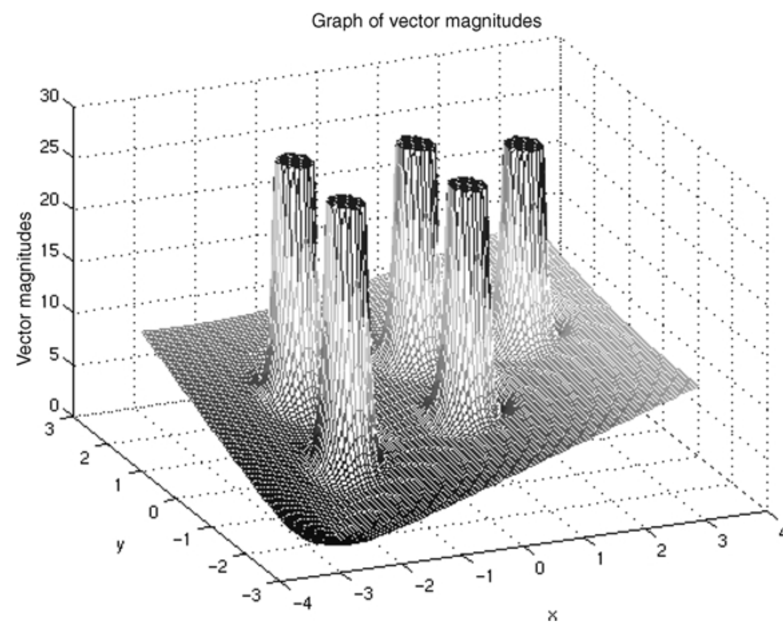


[Barranquand, Langlois, Latombe, 1989]

Comparison to attractor approach

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

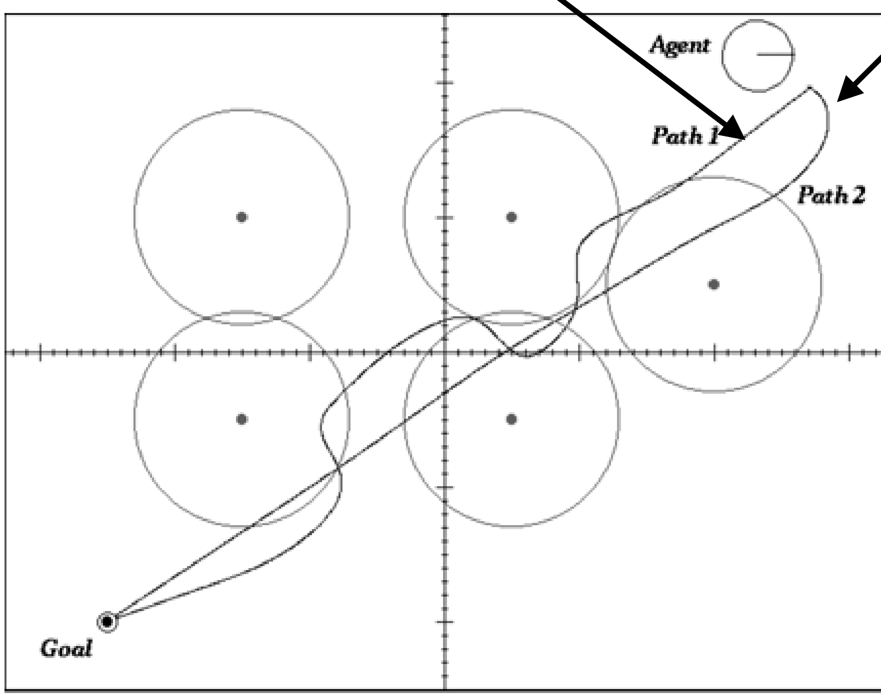


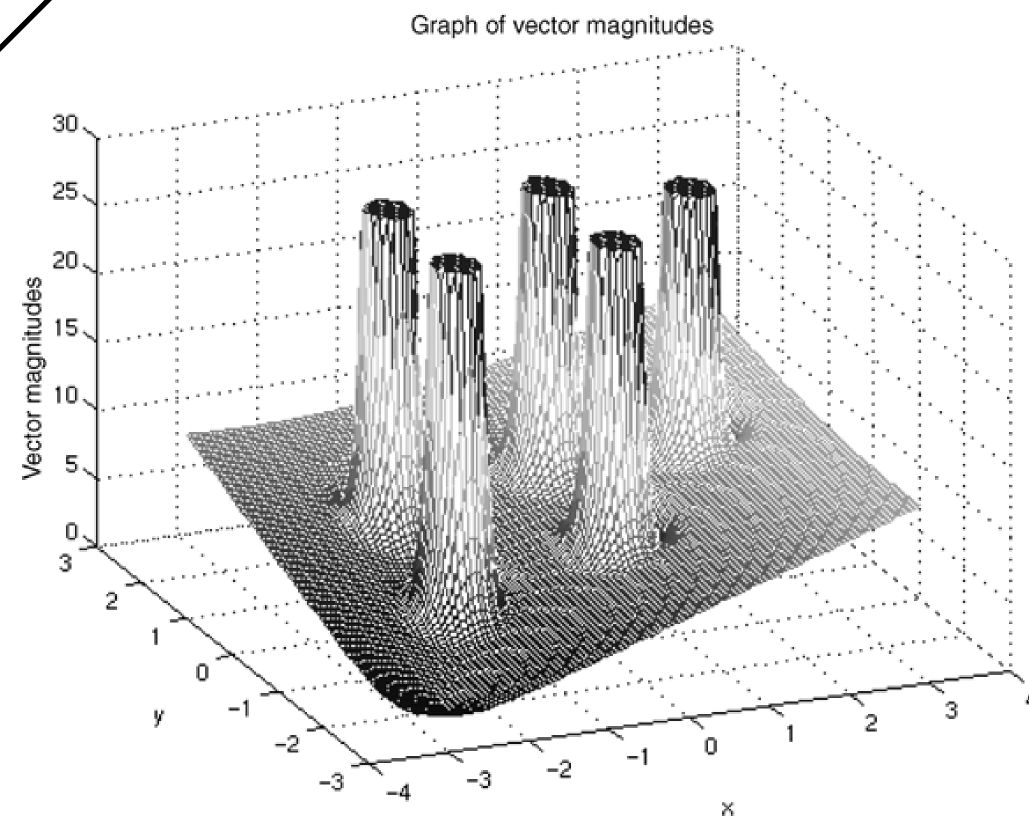


Typical paths of fitted models

potential field

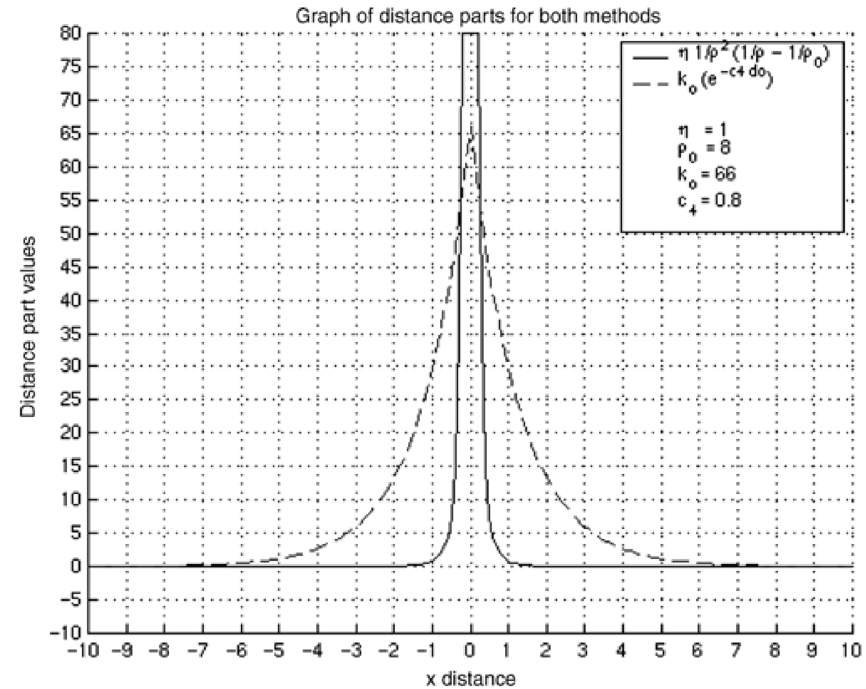
attractor dynamics



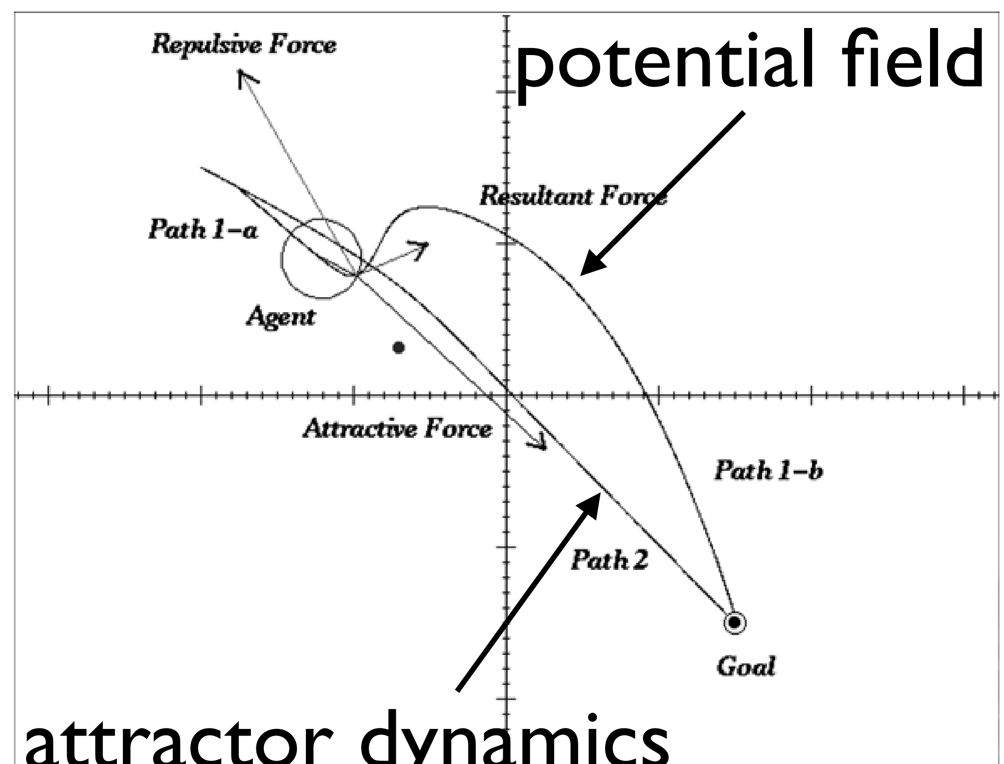


comparison potential field vs. attractor dynamics

- if potential is sharper than distance dependence of repellor: path too strongly curved



(a)

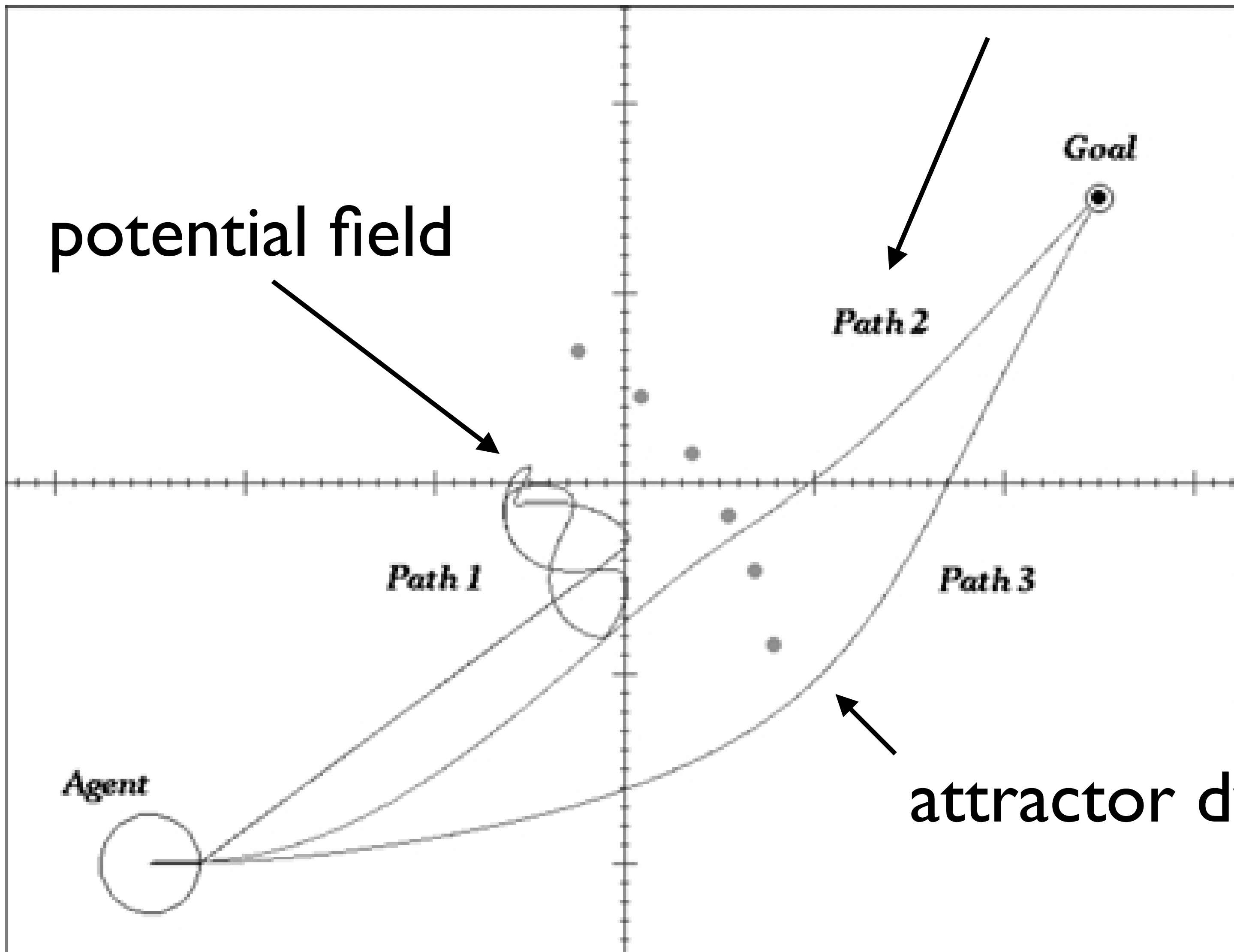


(b)

local minima vs. spurious attractor

attractor dynamics:
spurious attractor

potential field



attractor dynamics

local minima vs. spurious attractor

- in potential field approach, local minimal and constraint violations occur
- solution: make potential field approach exact and global by computing the potential to guarantee constraint satisfaction…
- => navigation functions that have the exact required maxima and minima (Rimon, Koditschek, 1992)
- but: computational costly, and requires global information

local minima vs. spurious attractor

- in attractor dynamics approach, spurious attractors come from cancellation of repulsive force-lets… analyzed in Dose, Schöner 1992
- solution proposed there: reduce number of contributions to avoid cancelation.. selecting only relevant contributions…

What is the conceptual difference between the potential field and the attractor dynamics approaches?

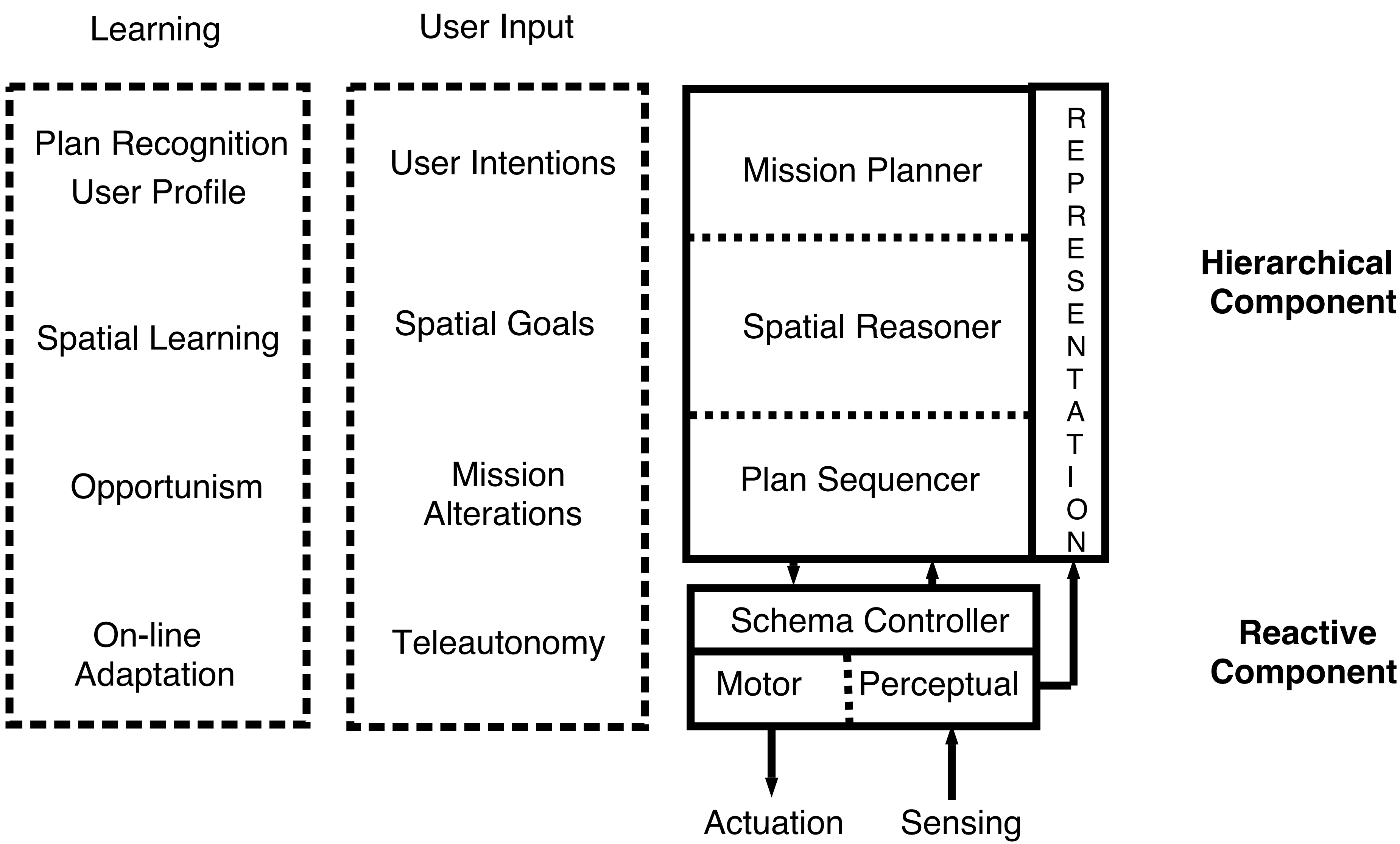
- => exercise

Potential fields as reactive planners

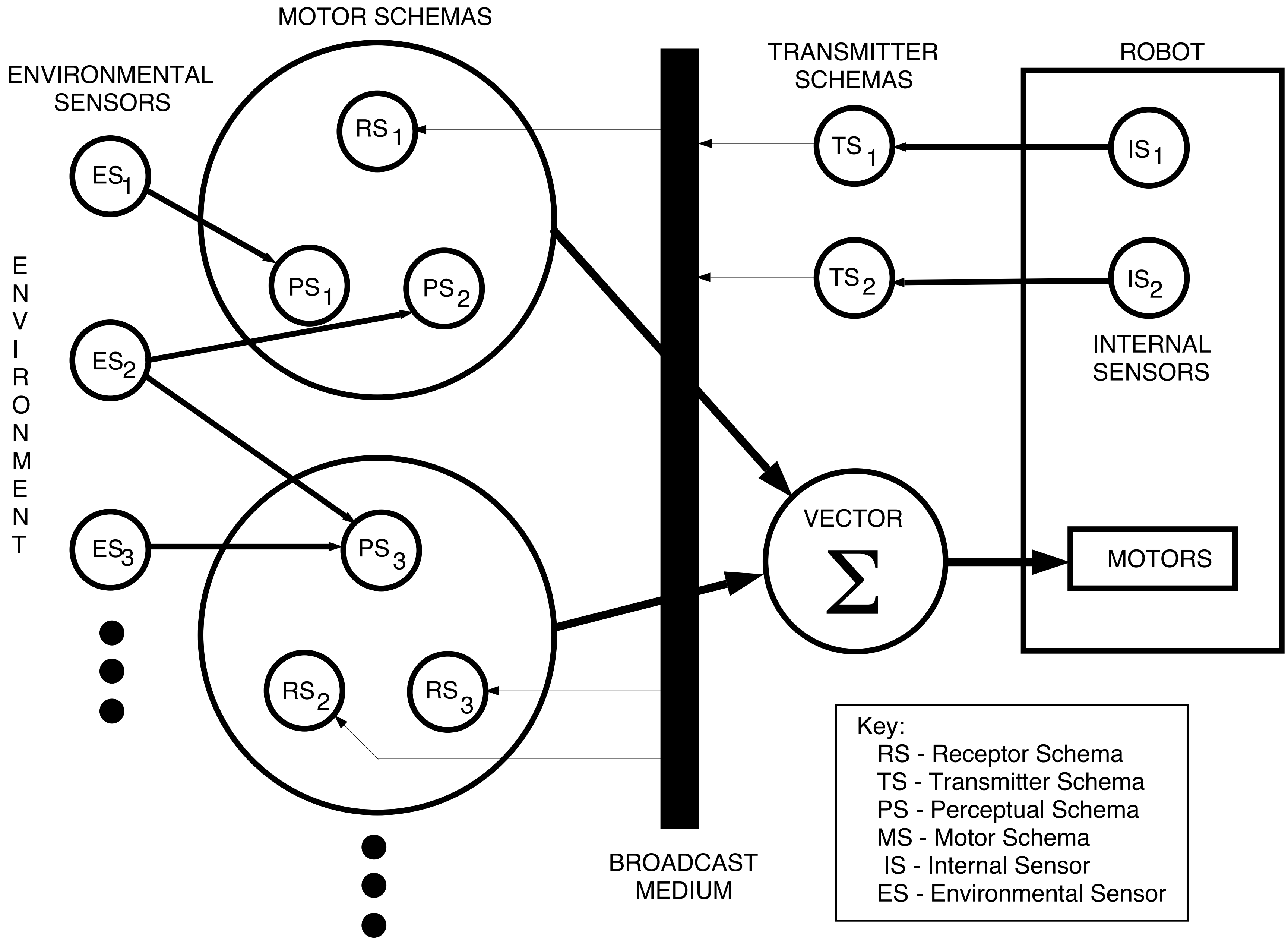
- use potential field to plan locally based on low-level sensory information (reactive)
- different "behaviors" generated by different vector-fields ("schema", slight generalization of potential fields)
- organize the different behaviors in an architecture

[Arkin, Blach: AuRA 1997]

Architecture



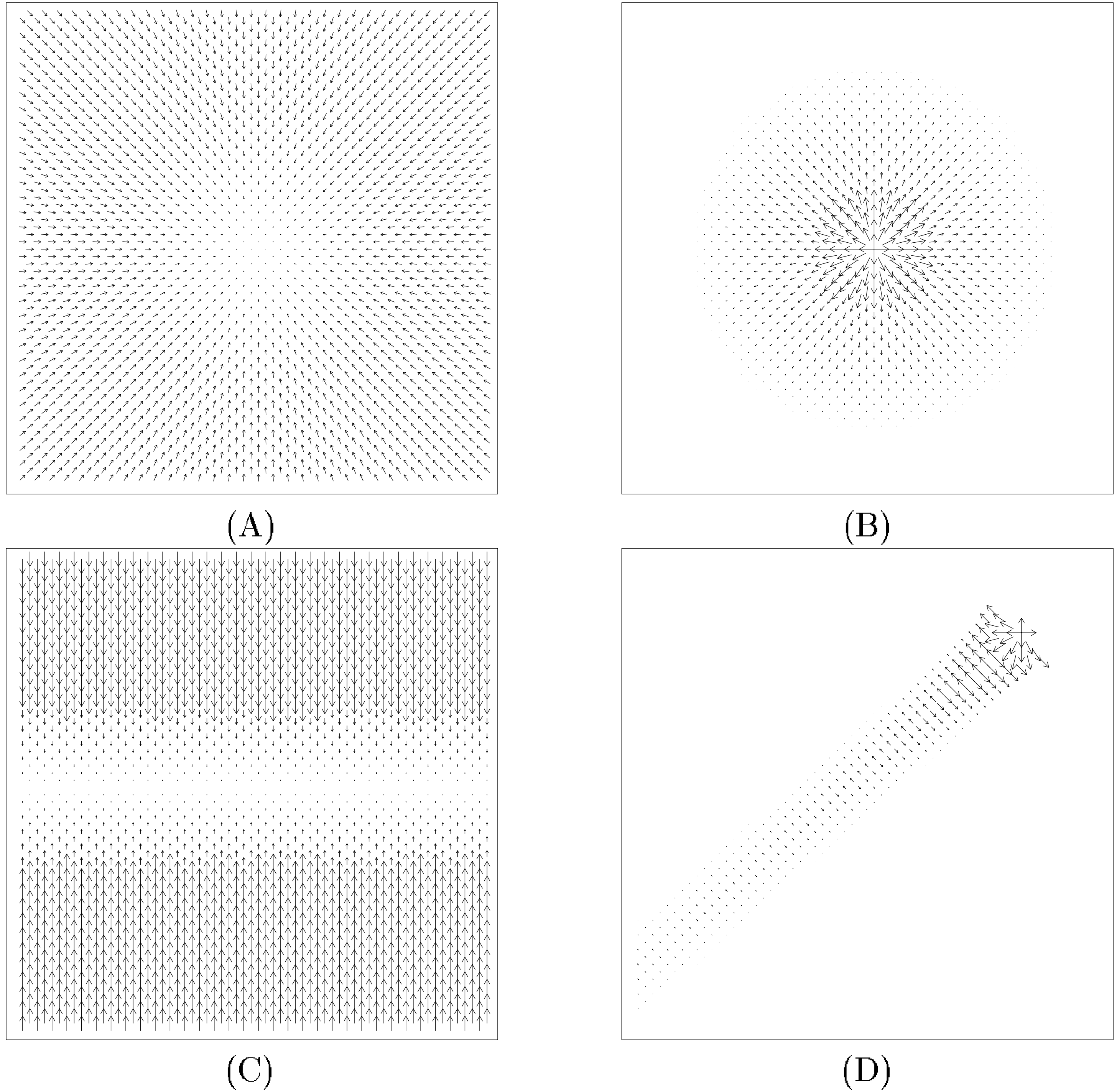
The reactive component



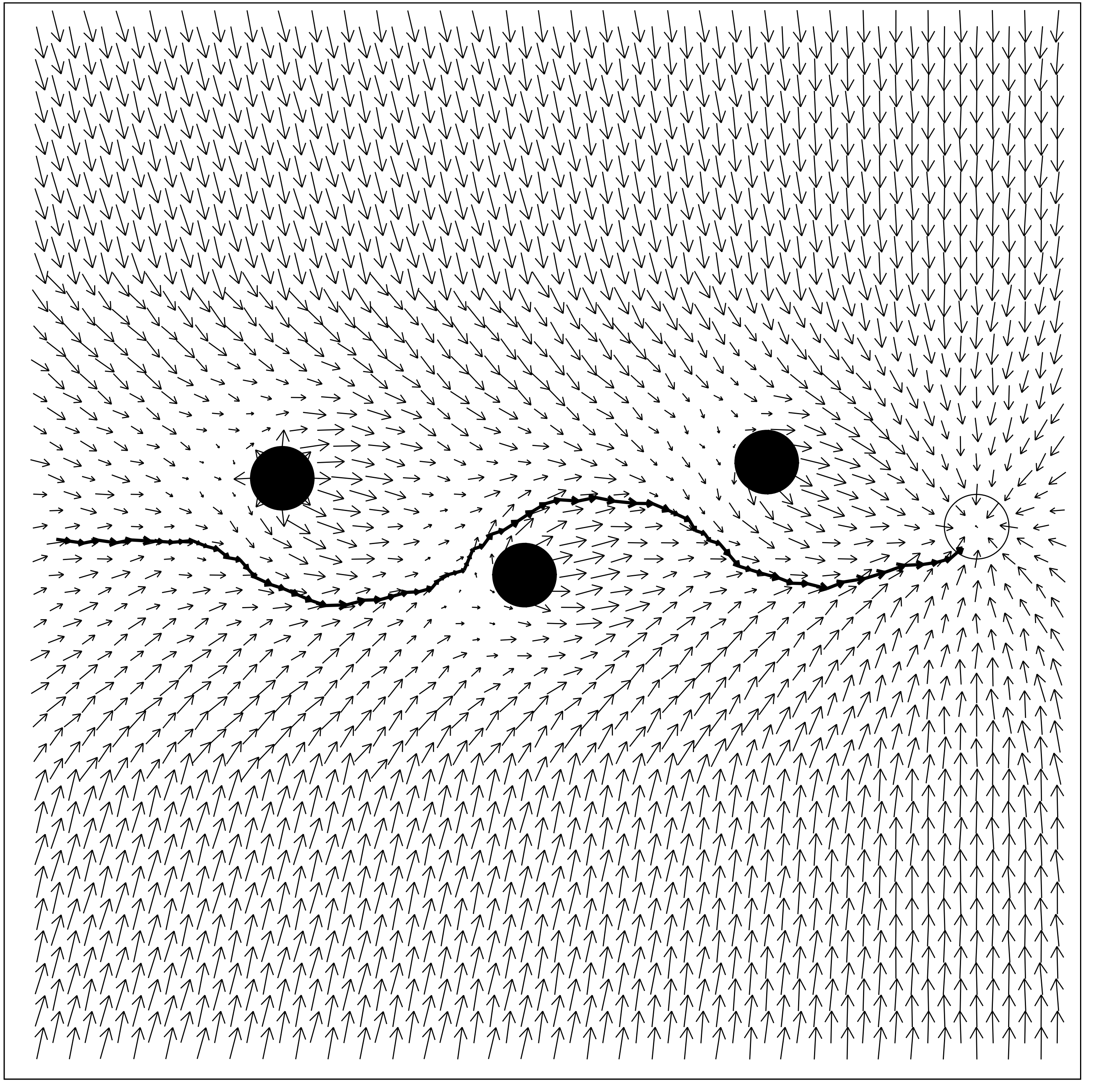
Motor schemata

- **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.

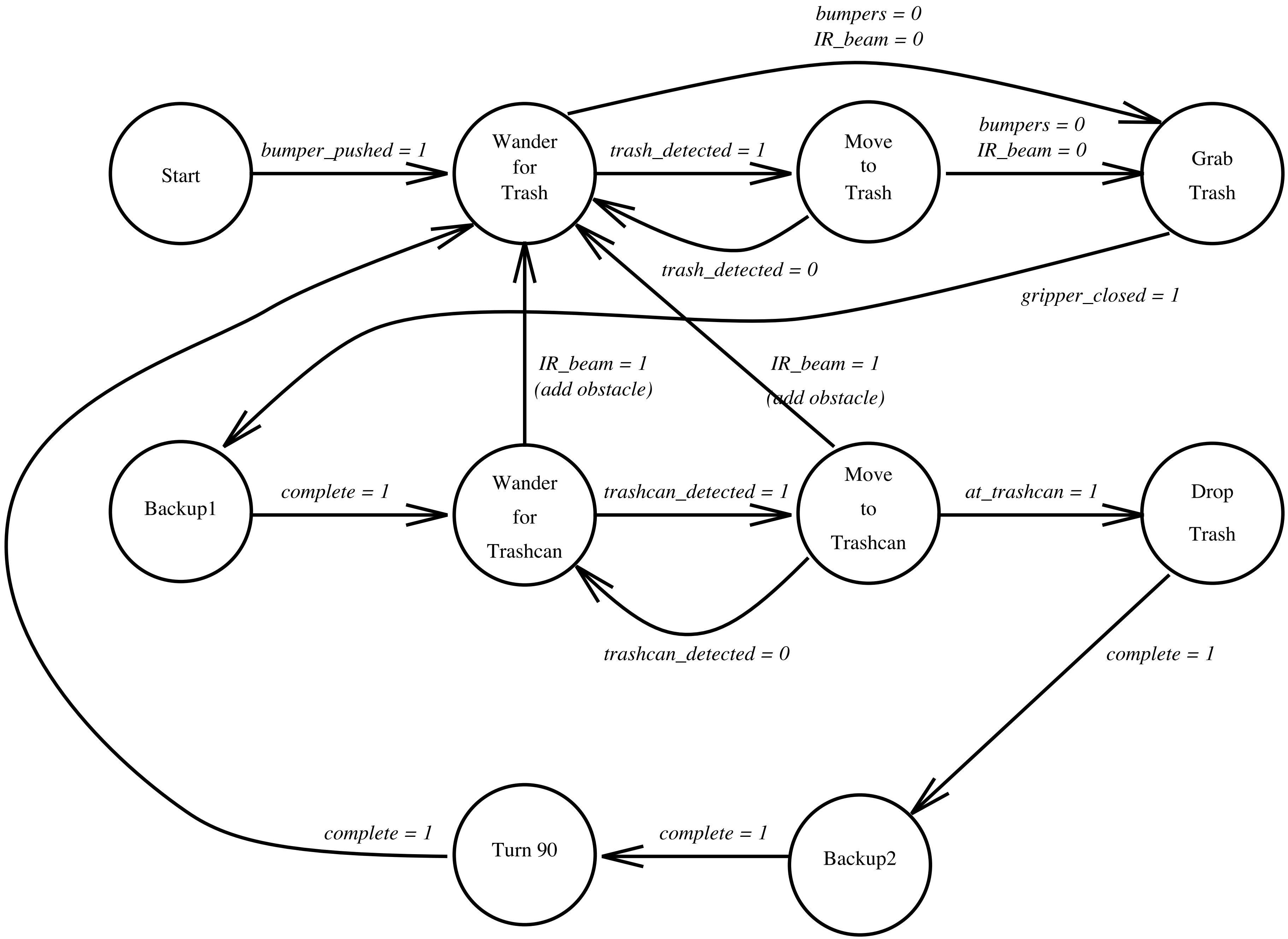
Vector-fields for different behaviors (schemata)



Superposing potential fields to combine behaviors

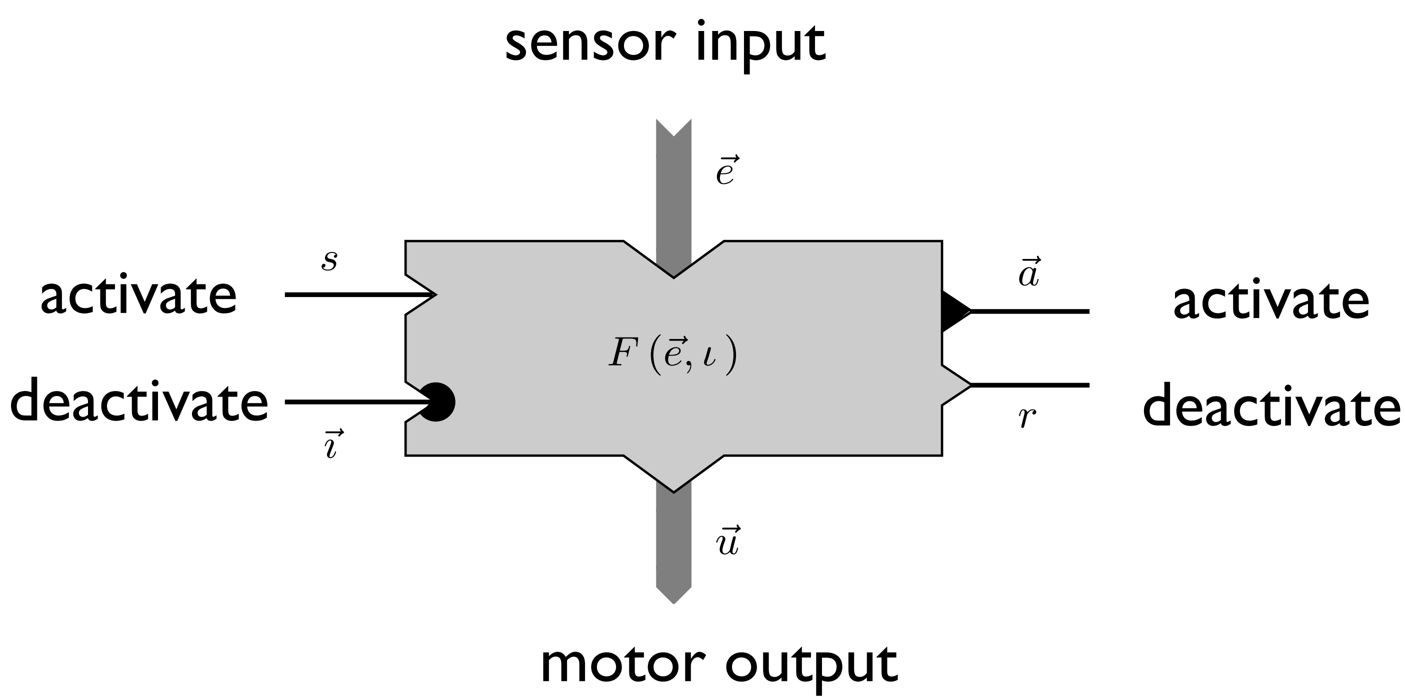


Behavior-based sequence planner



Scaling behavior-based architectures

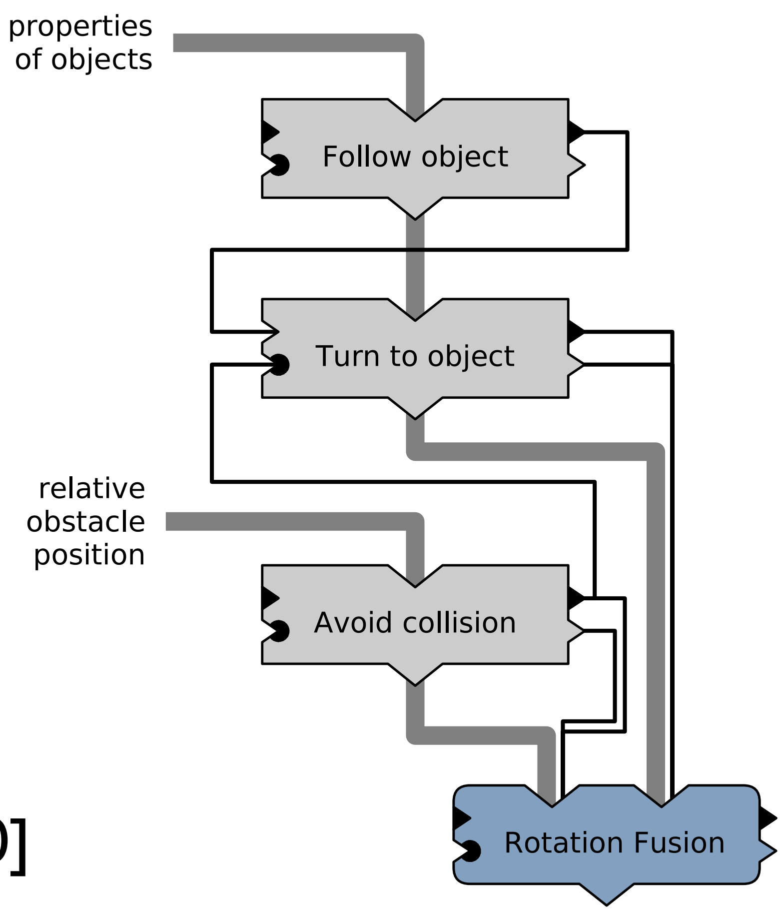
- behaviors: map sensor input to motor output
- are activated/deactivated
- and may in term activate/deactivate other behaviors



[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

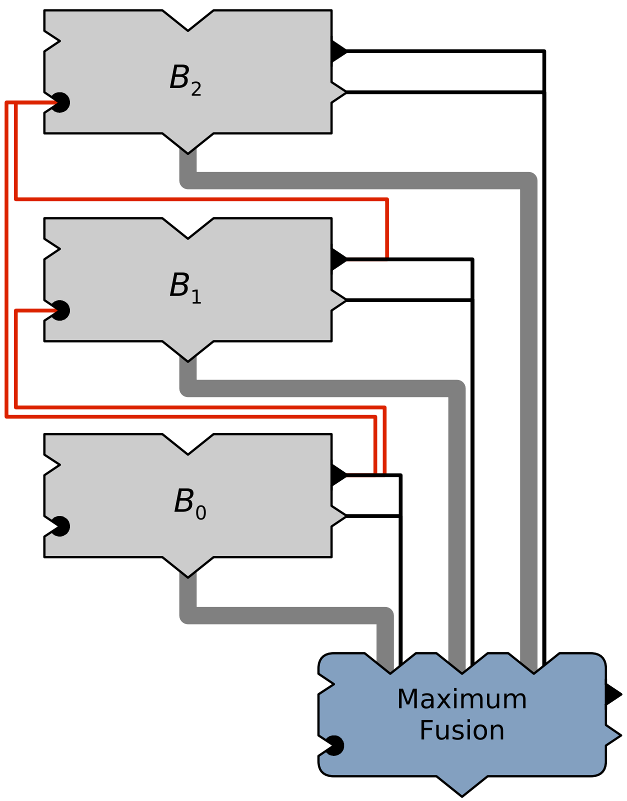


Fig. 5. Priority-based arbitration in iB2C.

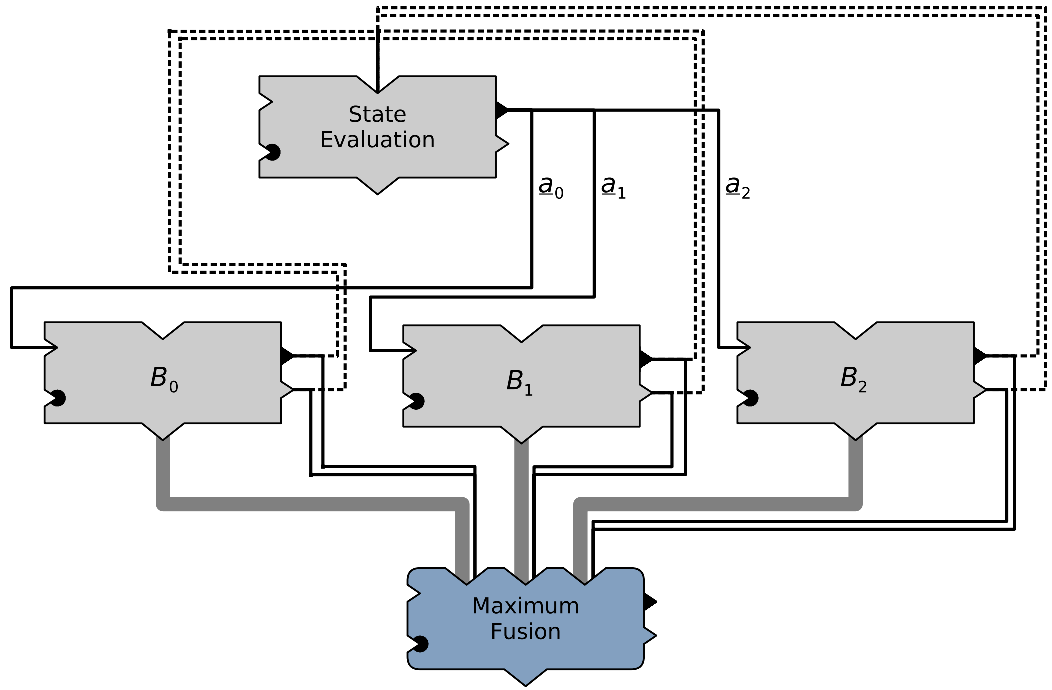


Fig. 6. State-based arbitration in iB2C.

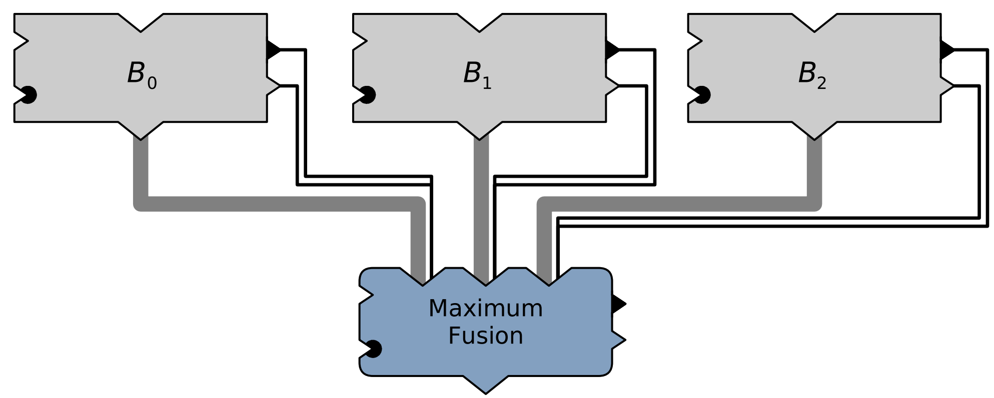
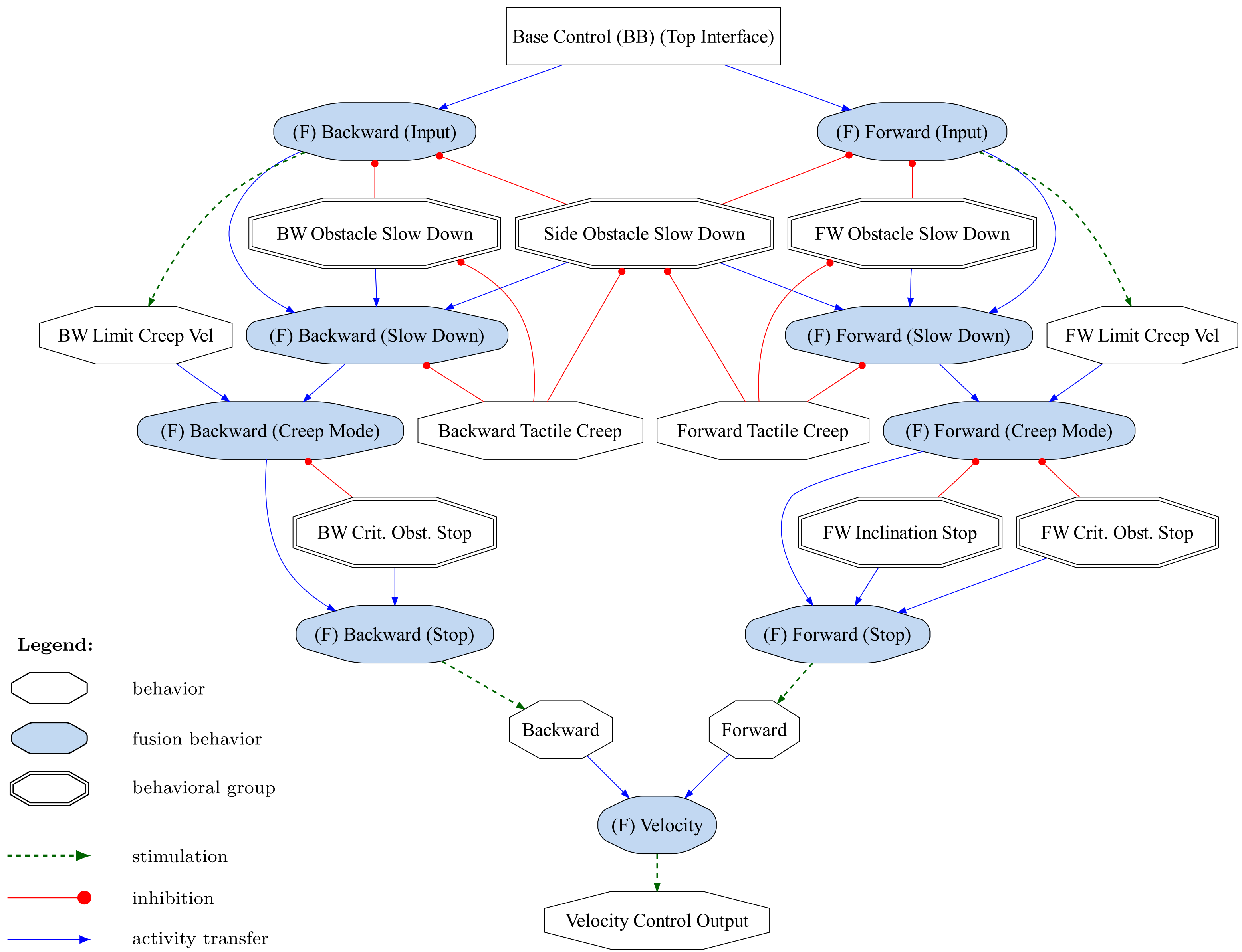


Fig. 7. Winner-takes-all arbitration in iB2C.

[Proetzsch, Luksch, Berns 2010]

Scaling behavior-based architectures



[Proetzsch, Luksch, Berns 2010]

Scaling behavior-based architectures

- implemented on a variety of systems

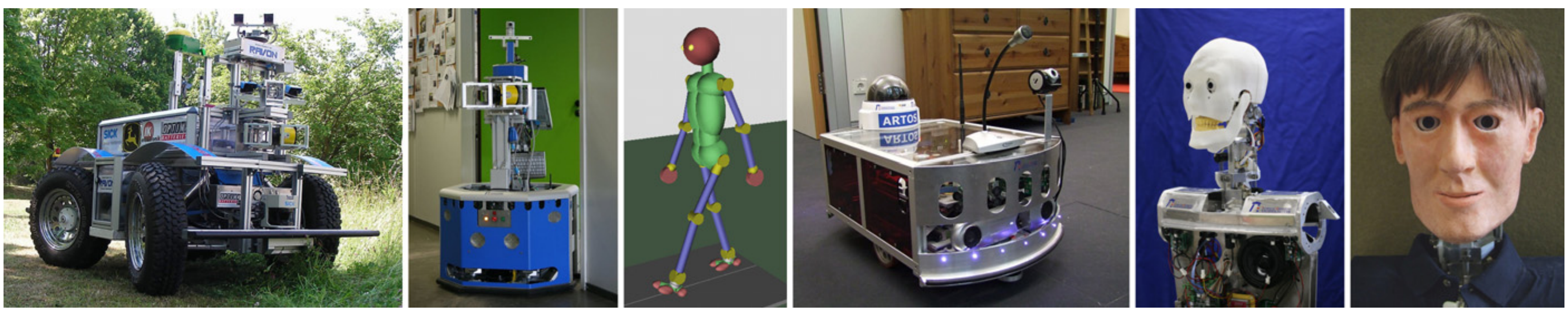
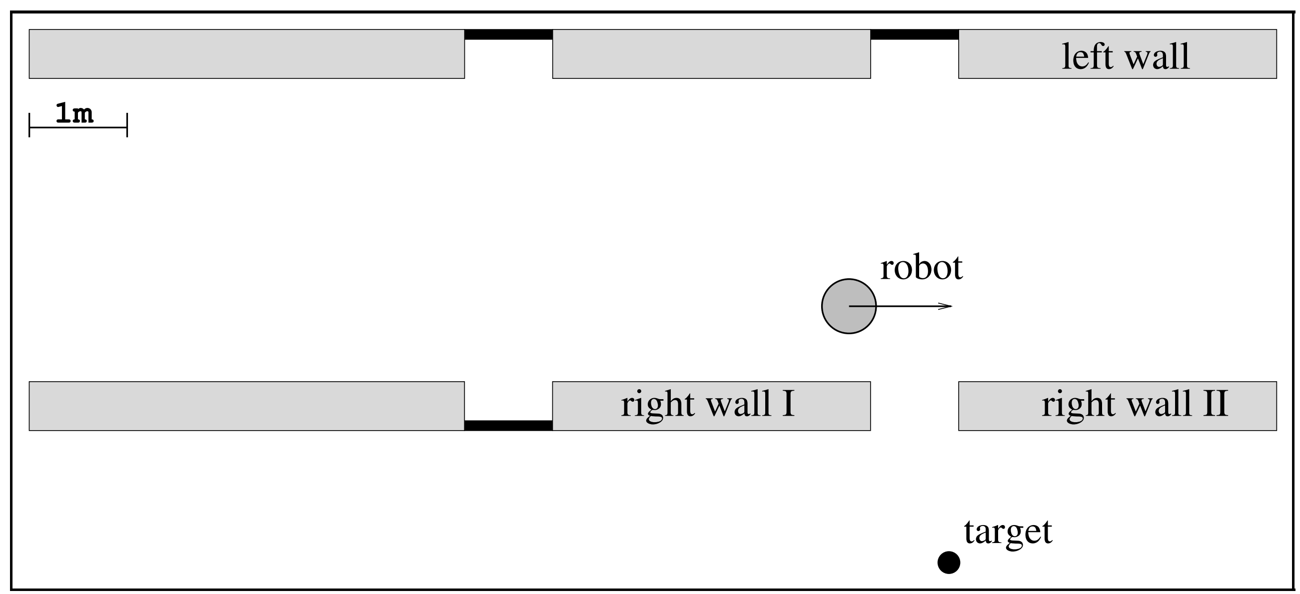


Fig. 20. Robots of the Robotics Research Lab controlled by an iB2C system: RAVON, MARVIN, dynamically simulated biped, ARTOS, and ROMAN (skeleton and skin).

[Proetzsch, Luksch, Berns 2010]

Dynamic window approach

- take dynamic constraints of vehicle into account (maximal decelerations/accelerations)… to drive fast
- in a dynamic variant of the potential field method



[Fox, Burghard, Thrun, 1996]

Dynamic window approach

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

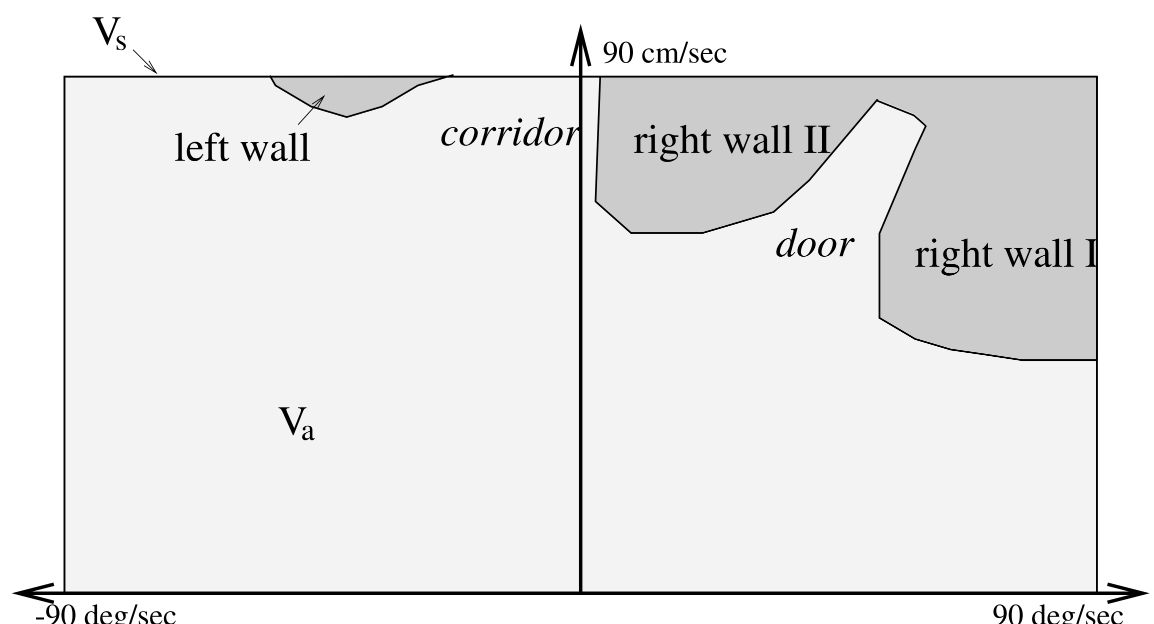


Figure 4. Velocity space

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

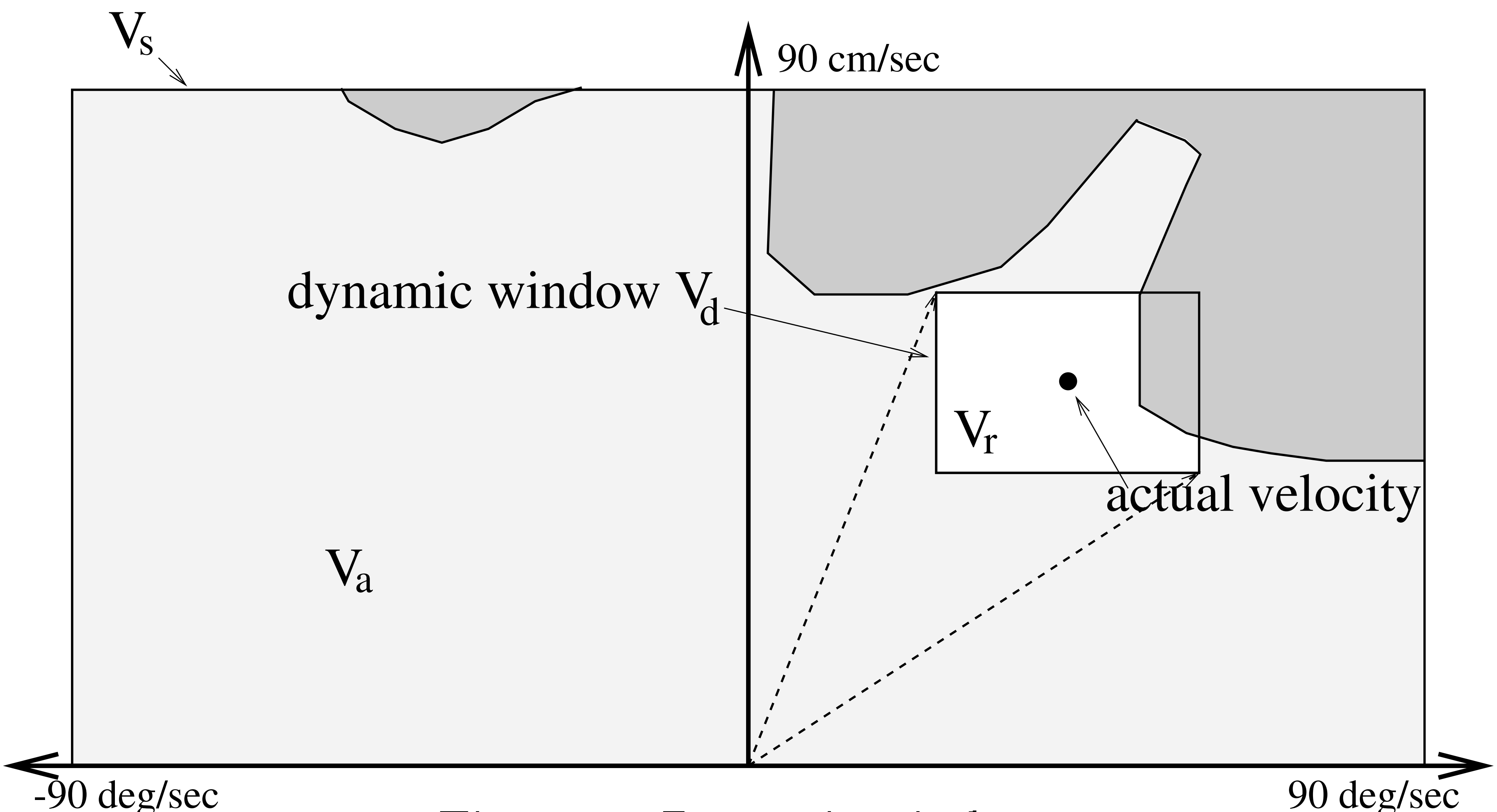


Figure 5. Dynamic window

Dynamic window approach

2. **Optimization:** The objective function

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

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

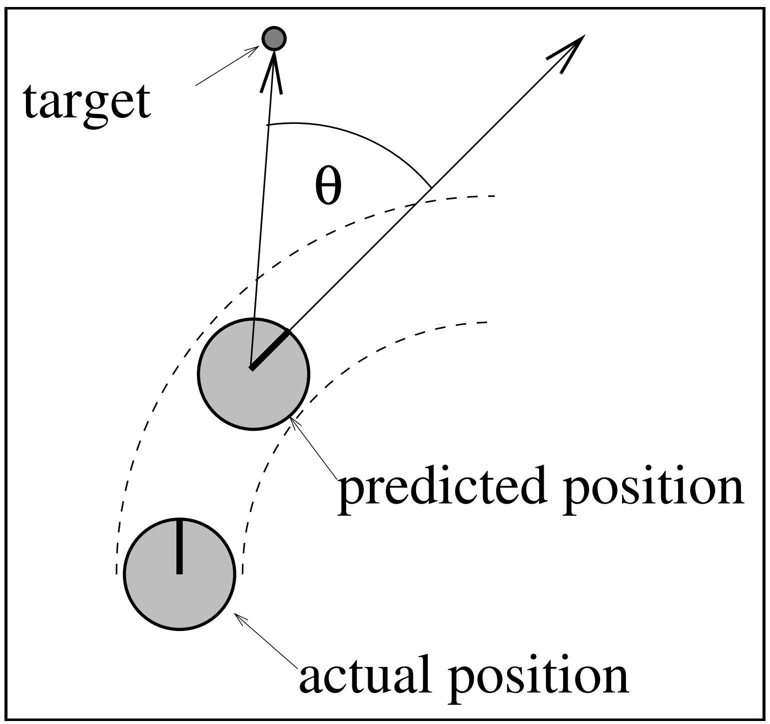


Figure 6. Angle θ to the target

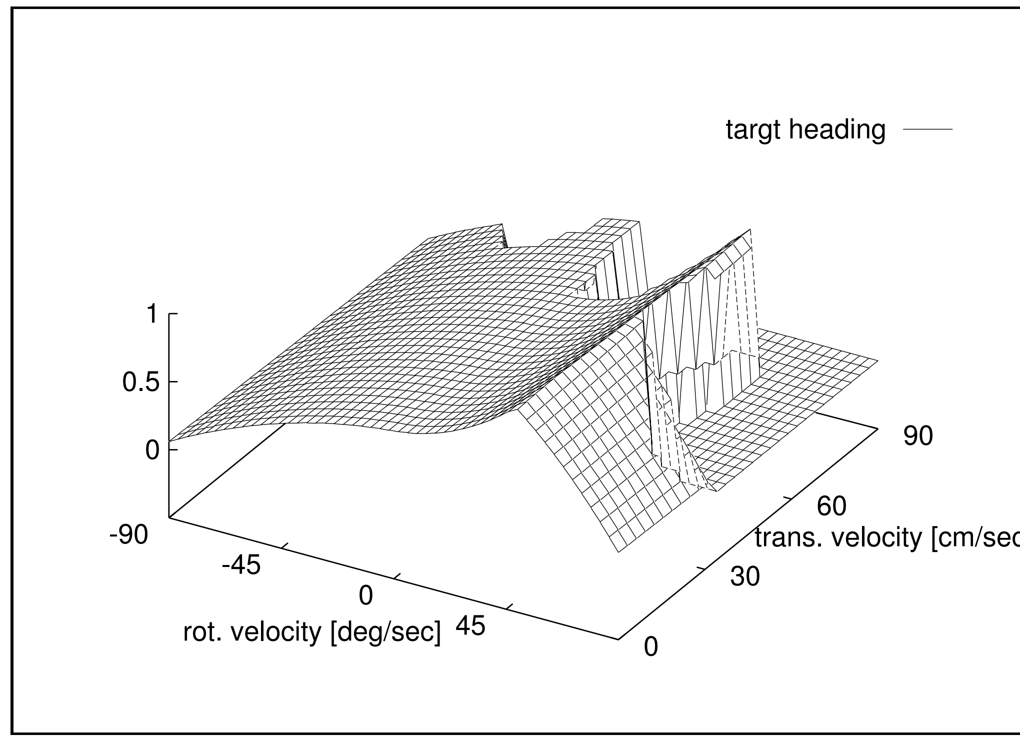


Figure 7. Evaluation of the target heading

Dynamic window approach

- clearance cost function

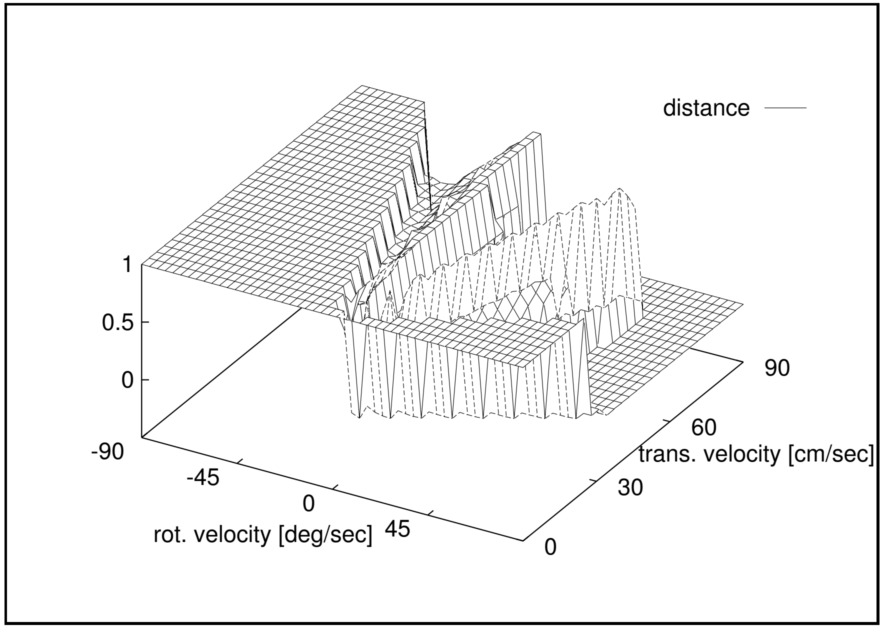


Figure 8. Evaluation of the distances

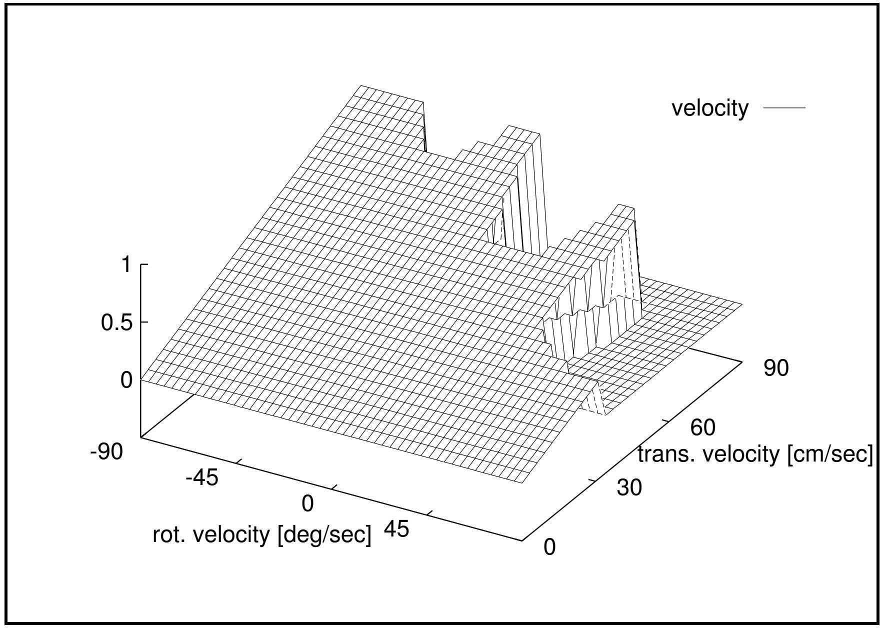


Figure 9. Evaluation of the velocities

Dynamic window approach

- smoothing the cost functions

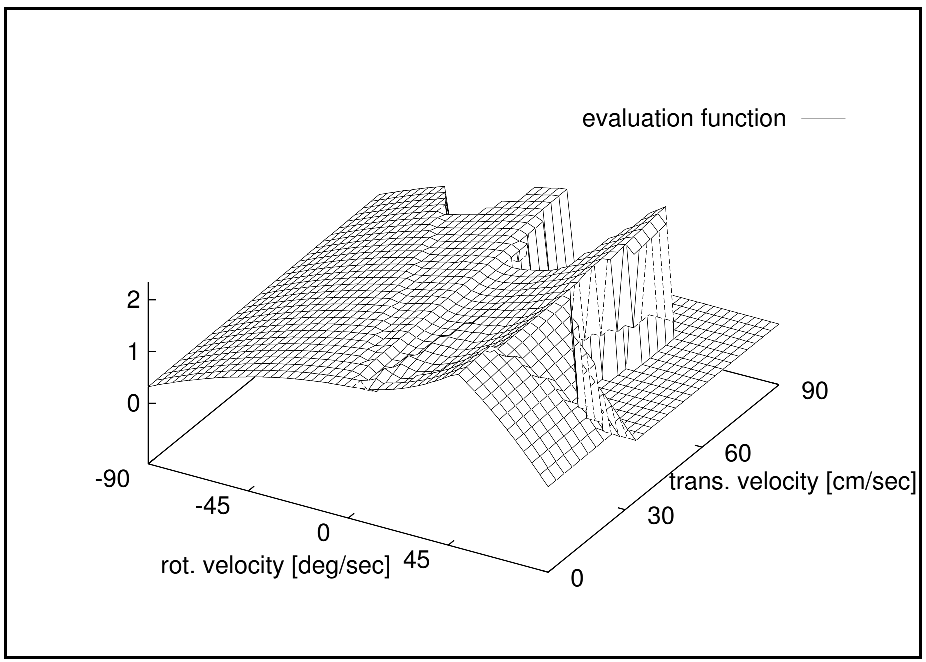


Figure 10. Combined evaluation function

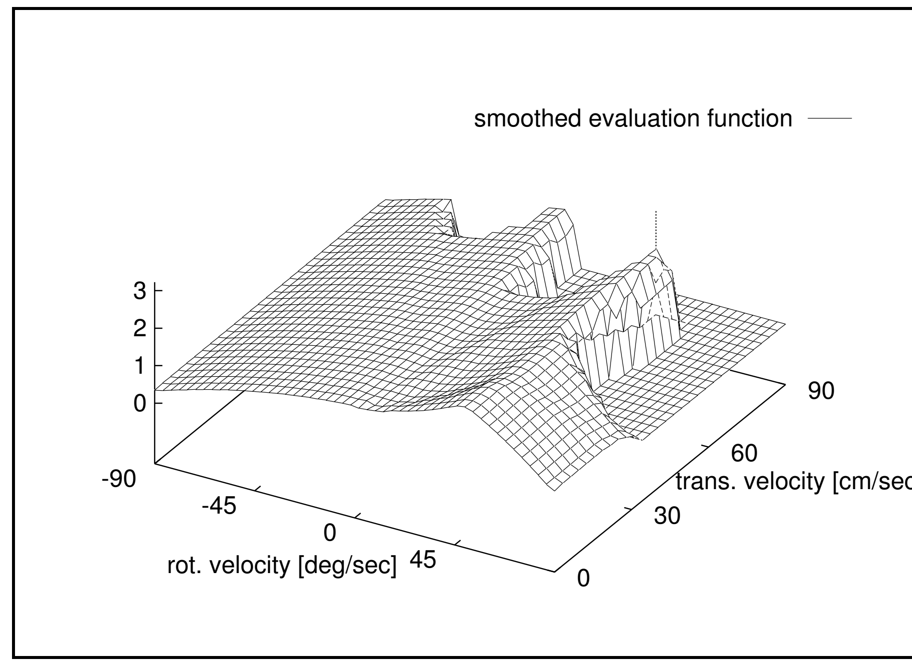


Figure 11. Objective function

Dynamic window approach

- two samples of actual velocities

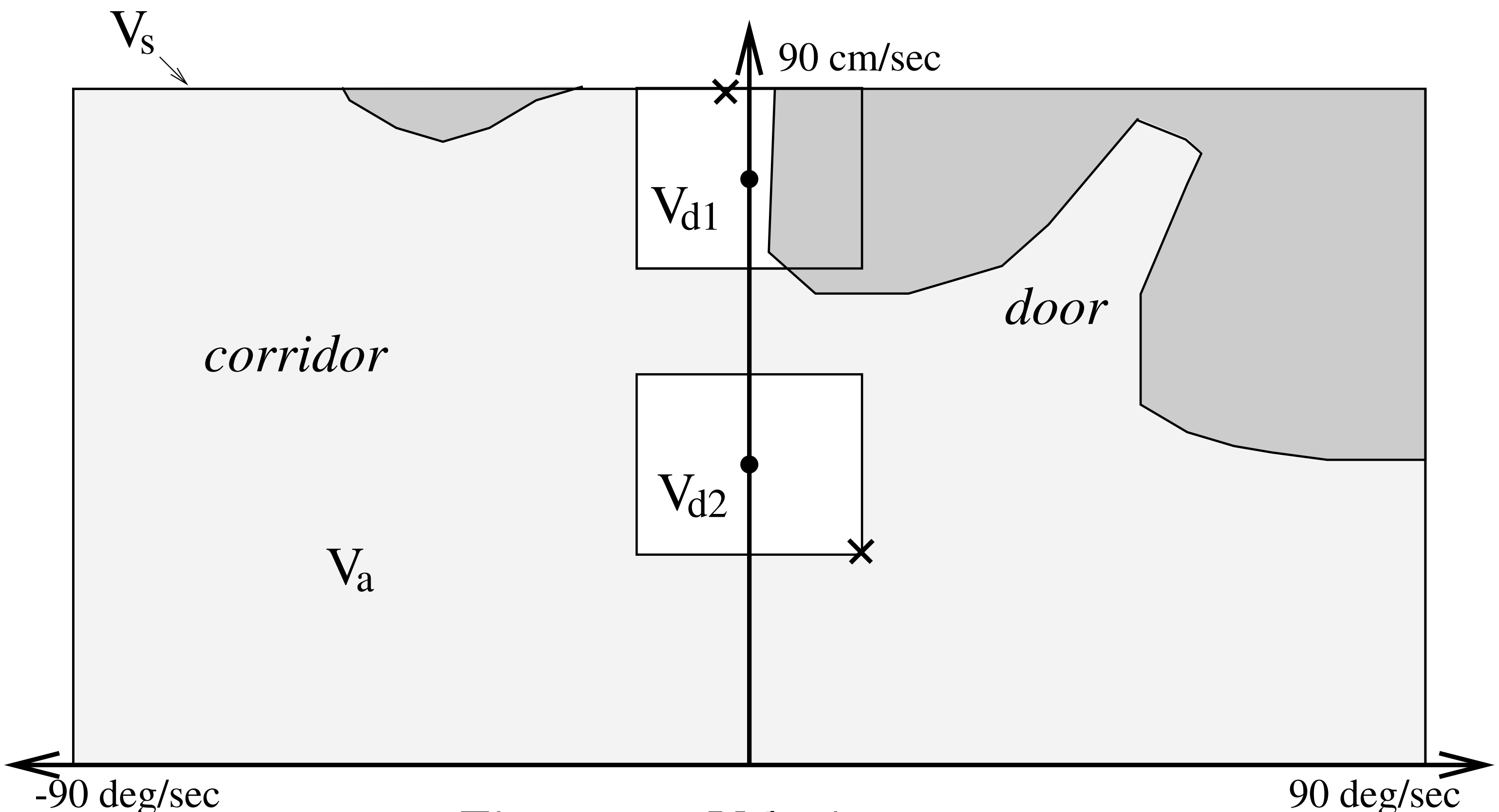


Figure 12. Velocity space

Dynamic window approach

- cost function for the action velocities

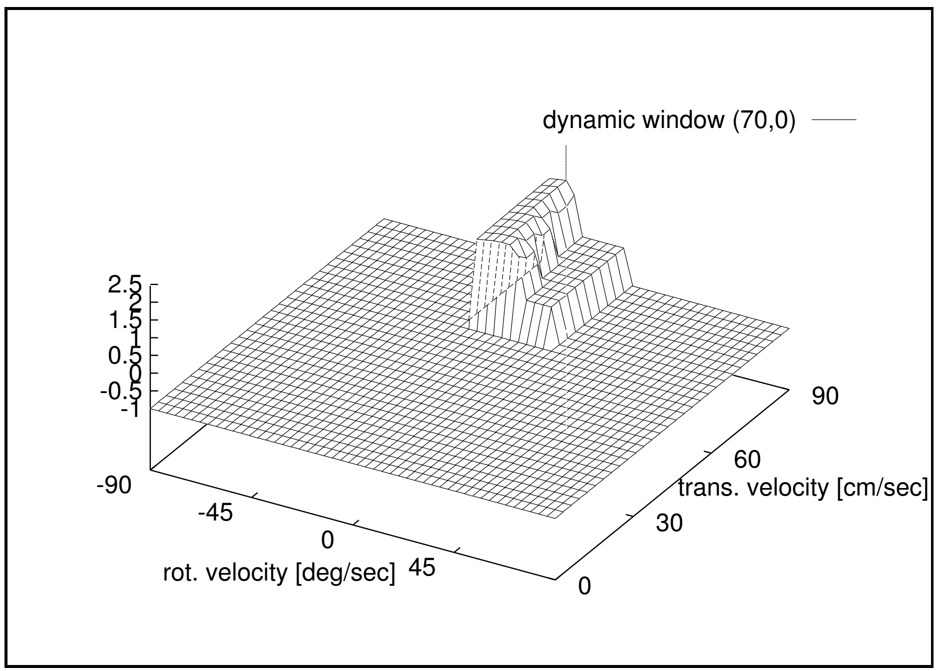


Figure 13. Objective function for actual velocity (75,0)

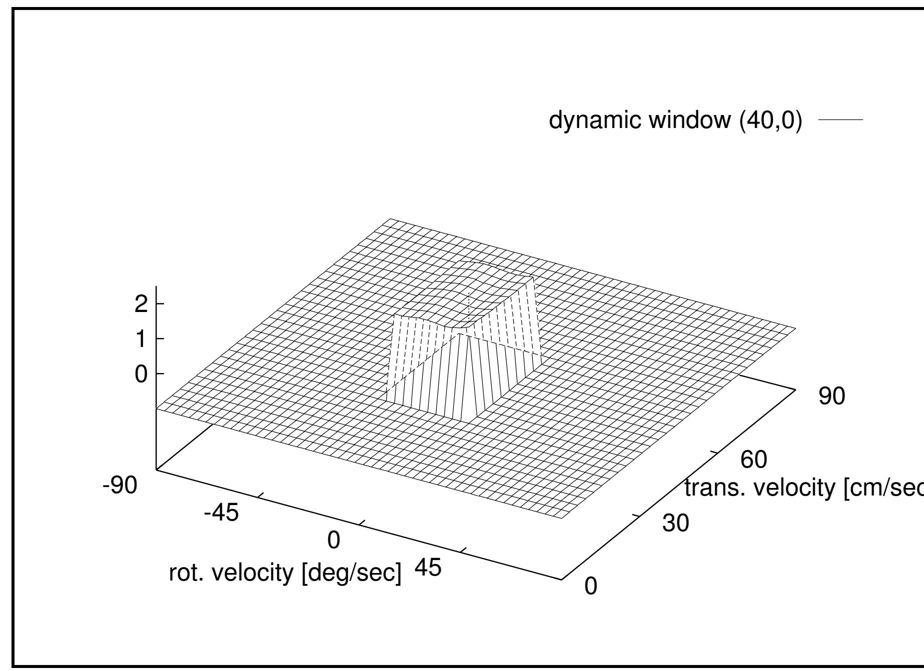


Figure 14. Objective function for actual velocity (40,0)

Dynamic window approach

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

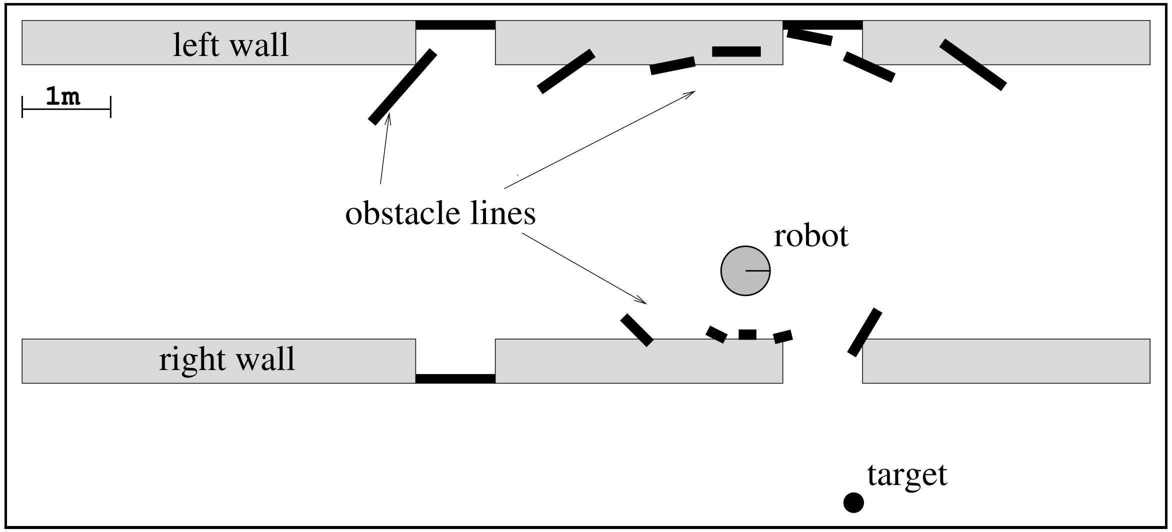


Figure 18. Example environment with obstacle lines and target point

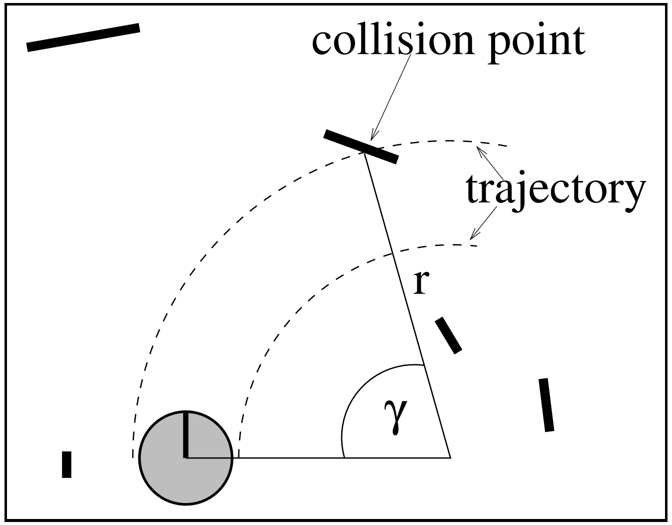


Figure 19. Determination of the distance

Dynamic window approach

- data

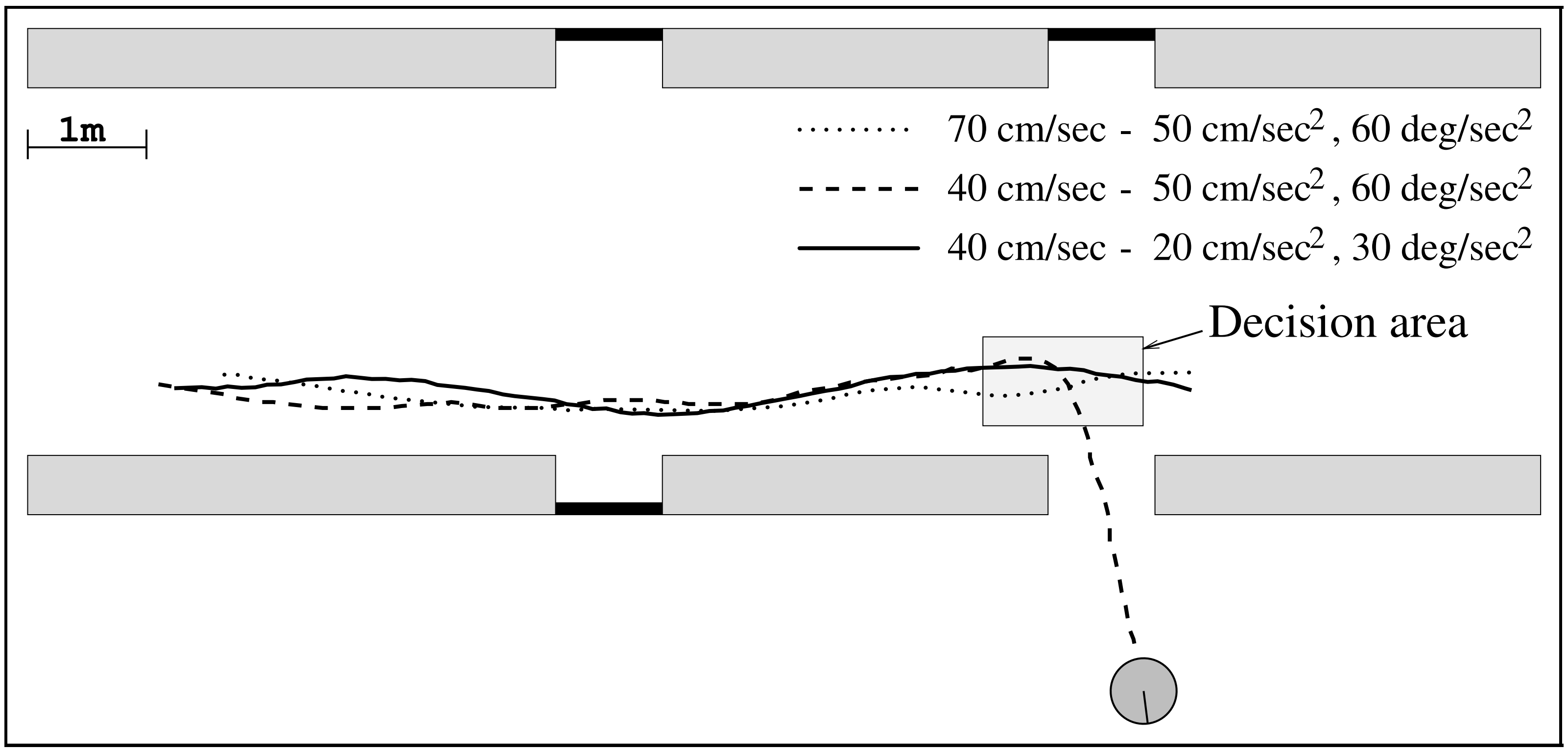


Figure 20. Trajectories chosen for different dynamic parameters

Dynamic window approach

- data

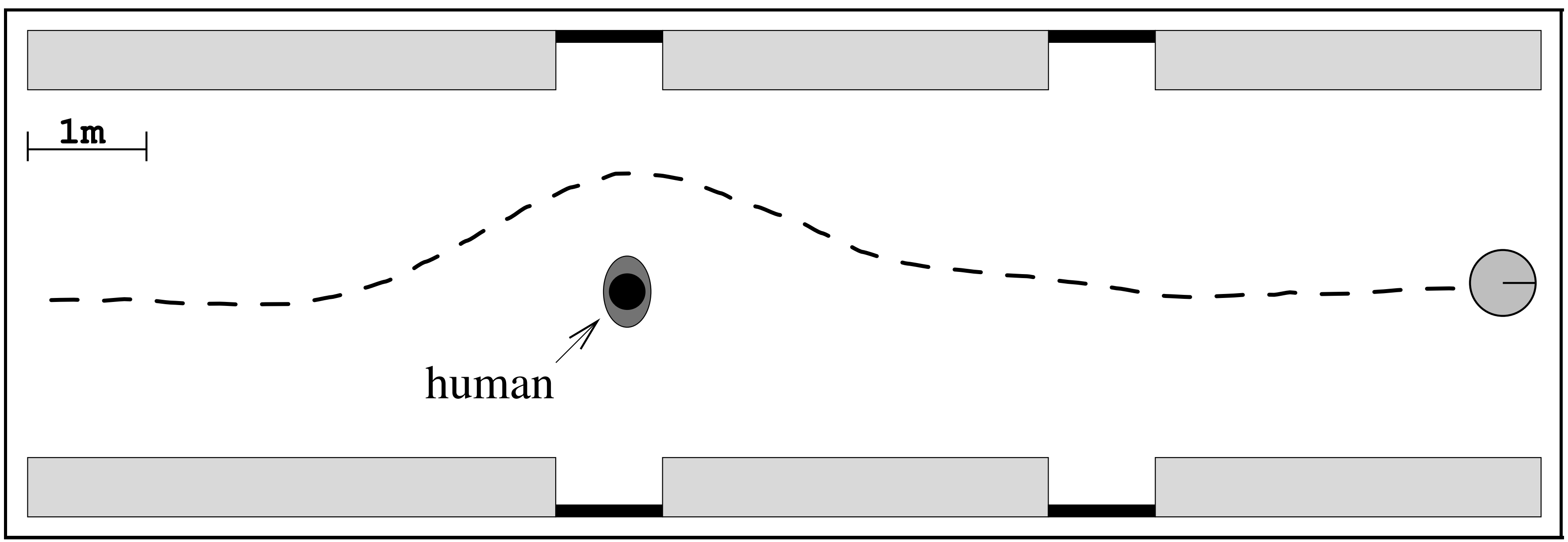


Figure 21. Trajectory through corridor

Dynamic window approach

- data

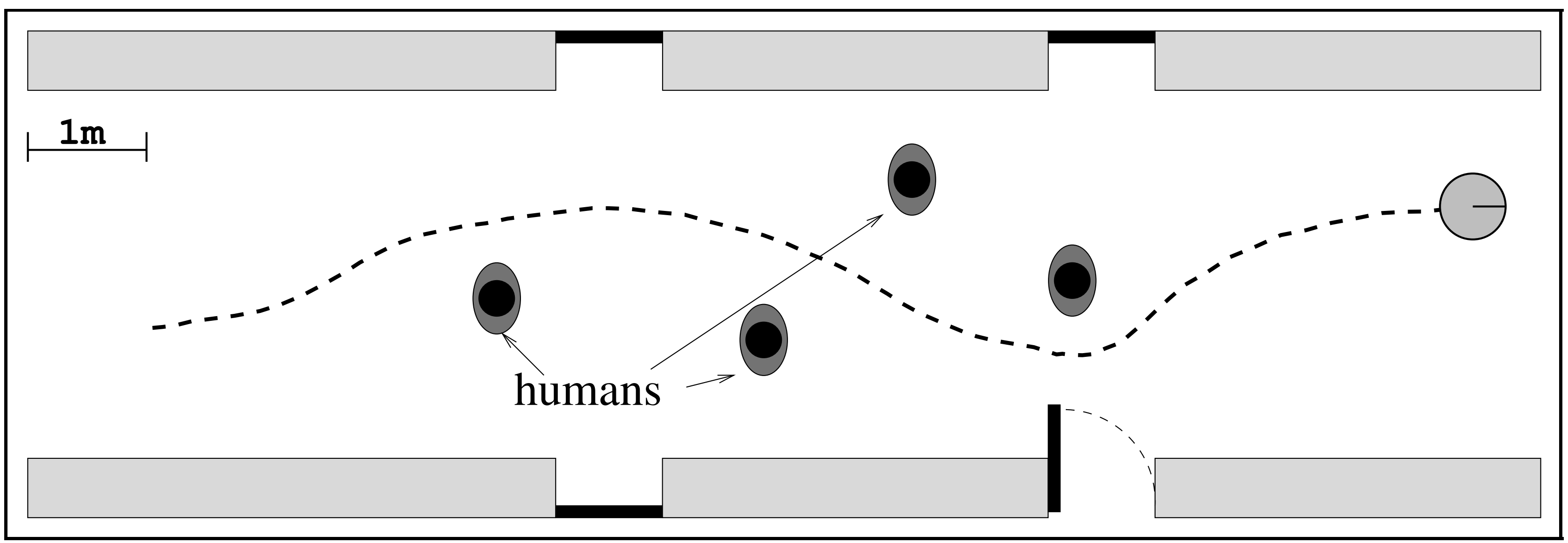


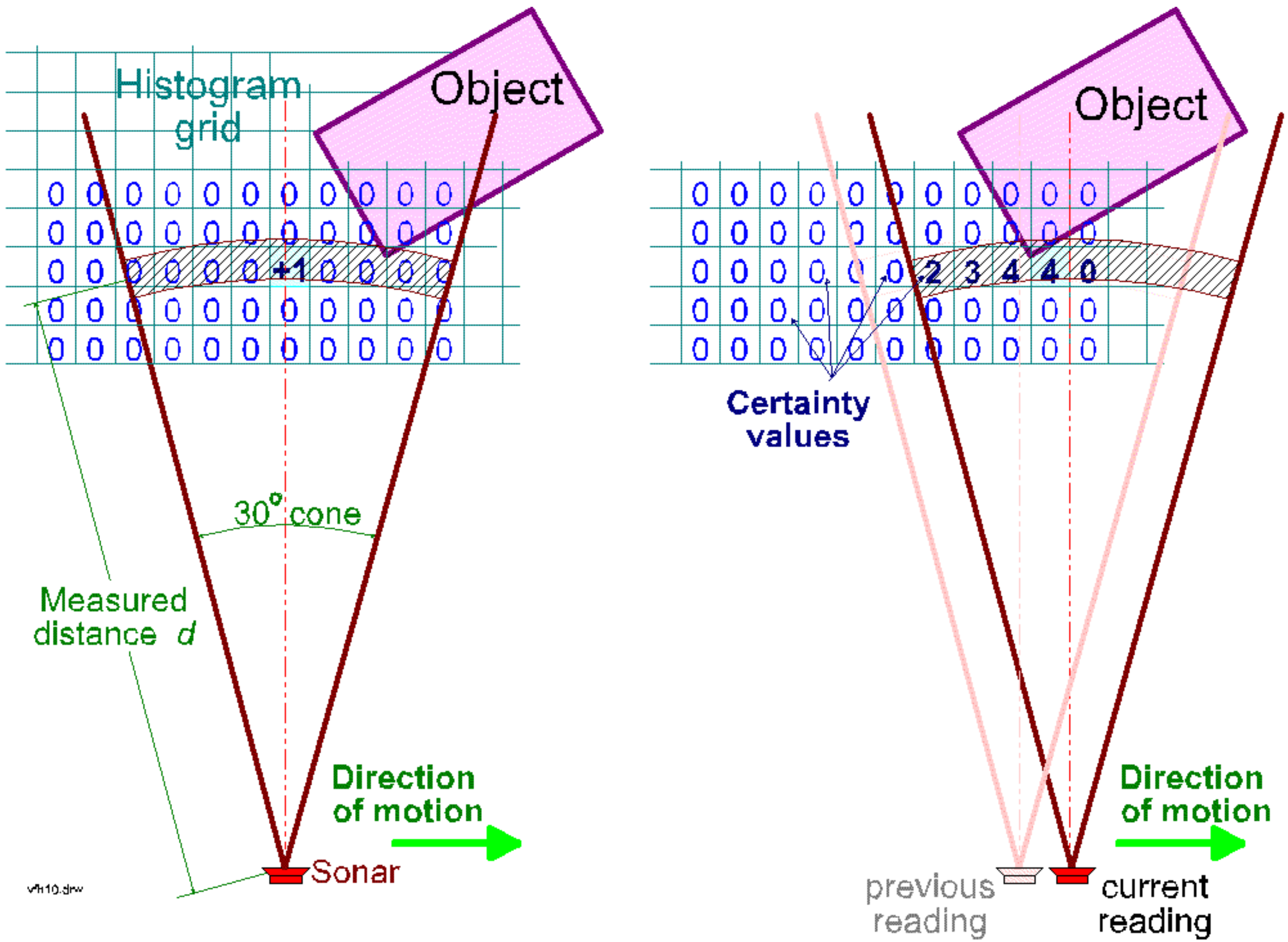
Figure 22. Trajectory through cluttered corridor

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

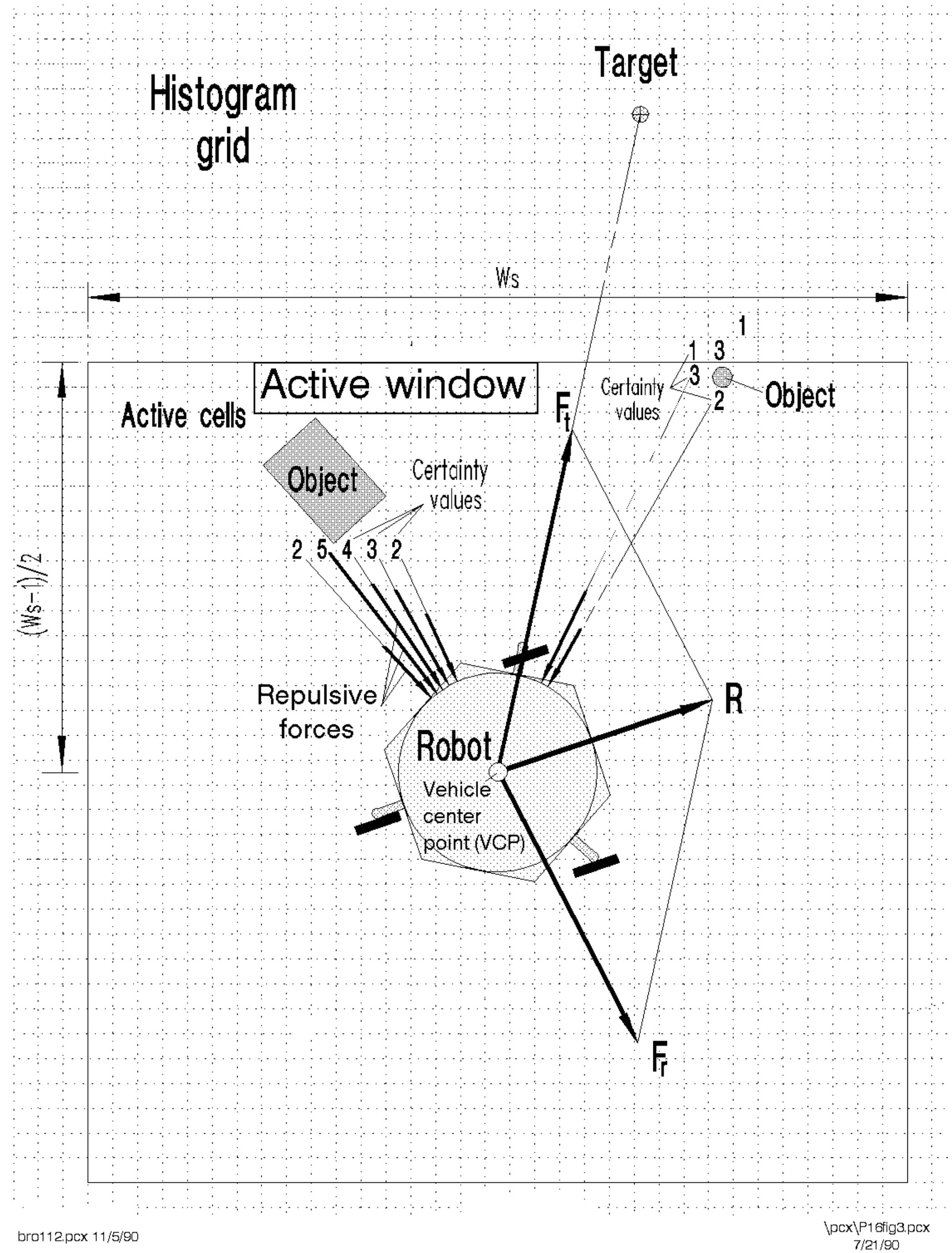
[Koren, Borenstein, 1991)

Virtual force field: Borenstein & Koren



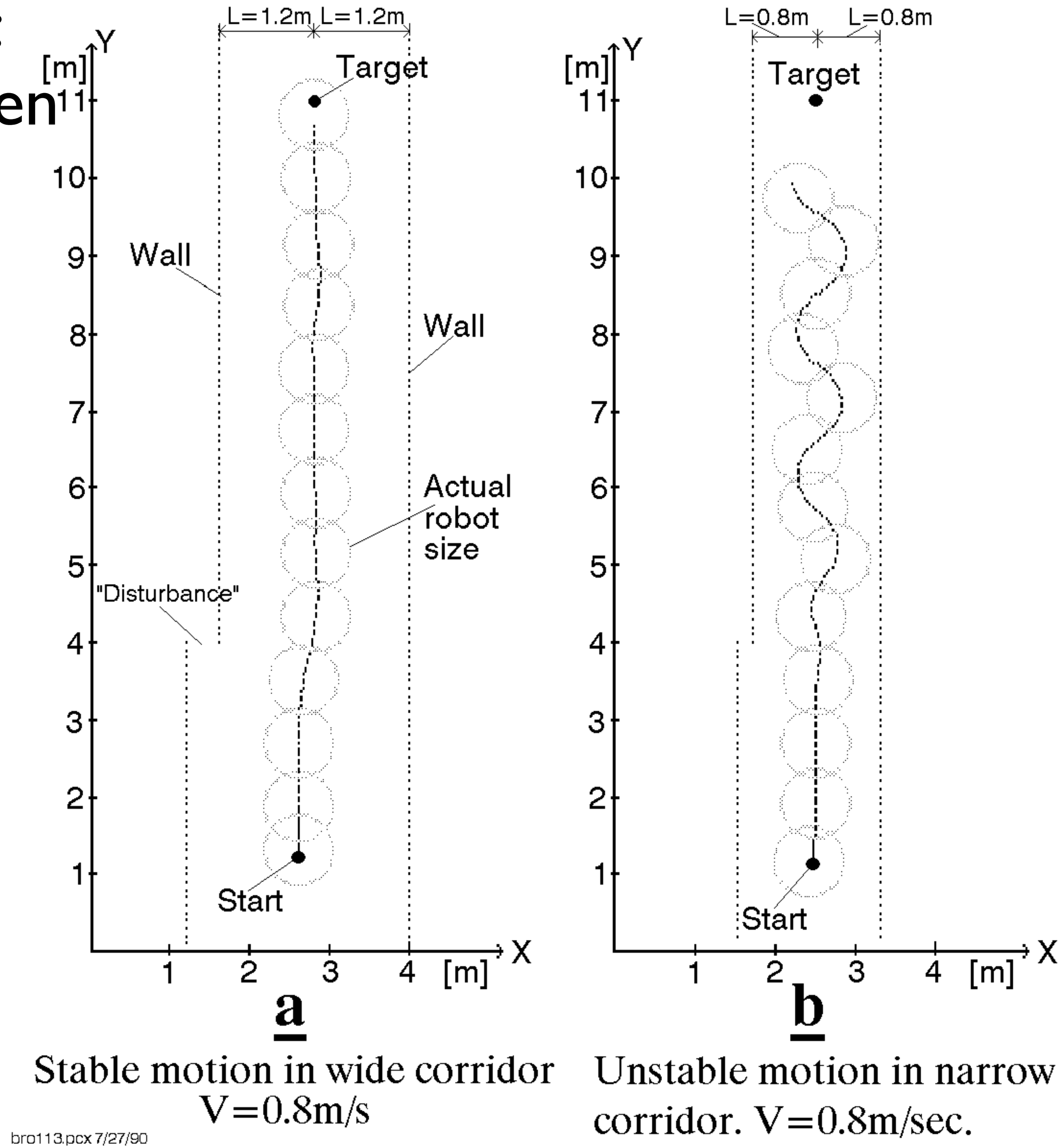
Virtual force field: Borenstein & Koren

- 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

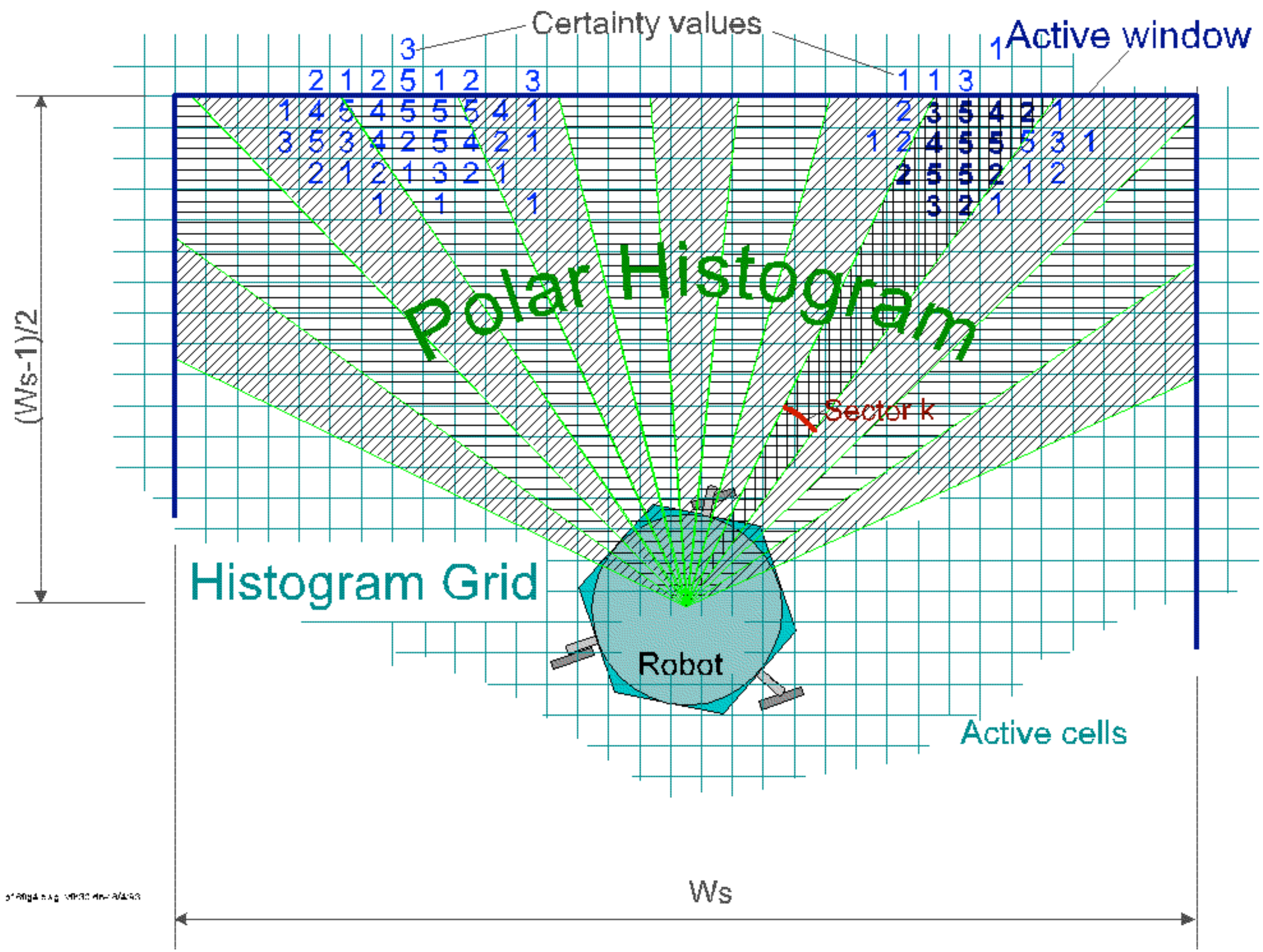


Stable motion in wide corridor V=0.8m/s

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

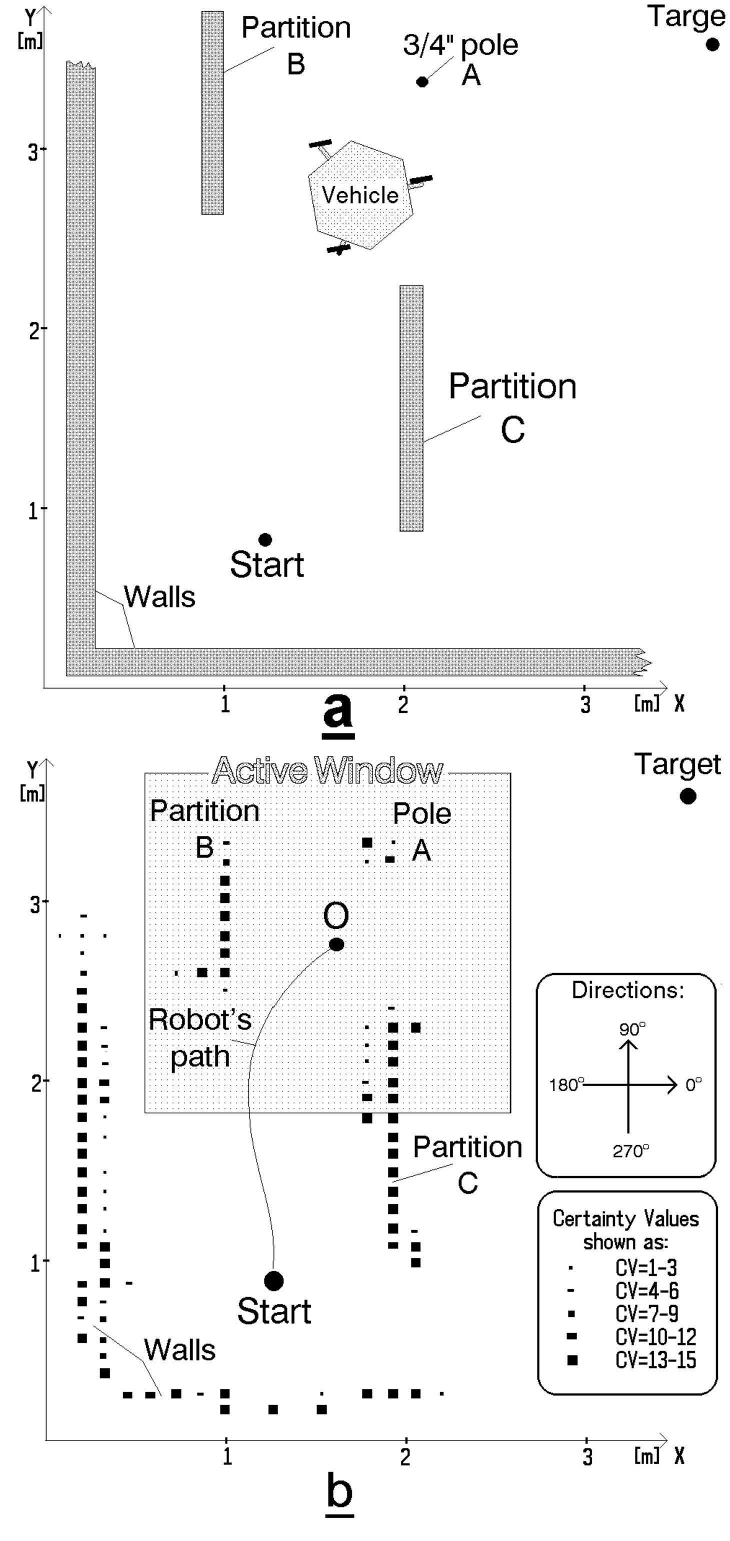
Vector field histogram: Borenstein & Koren

- transform active window in world grid into polar histogram



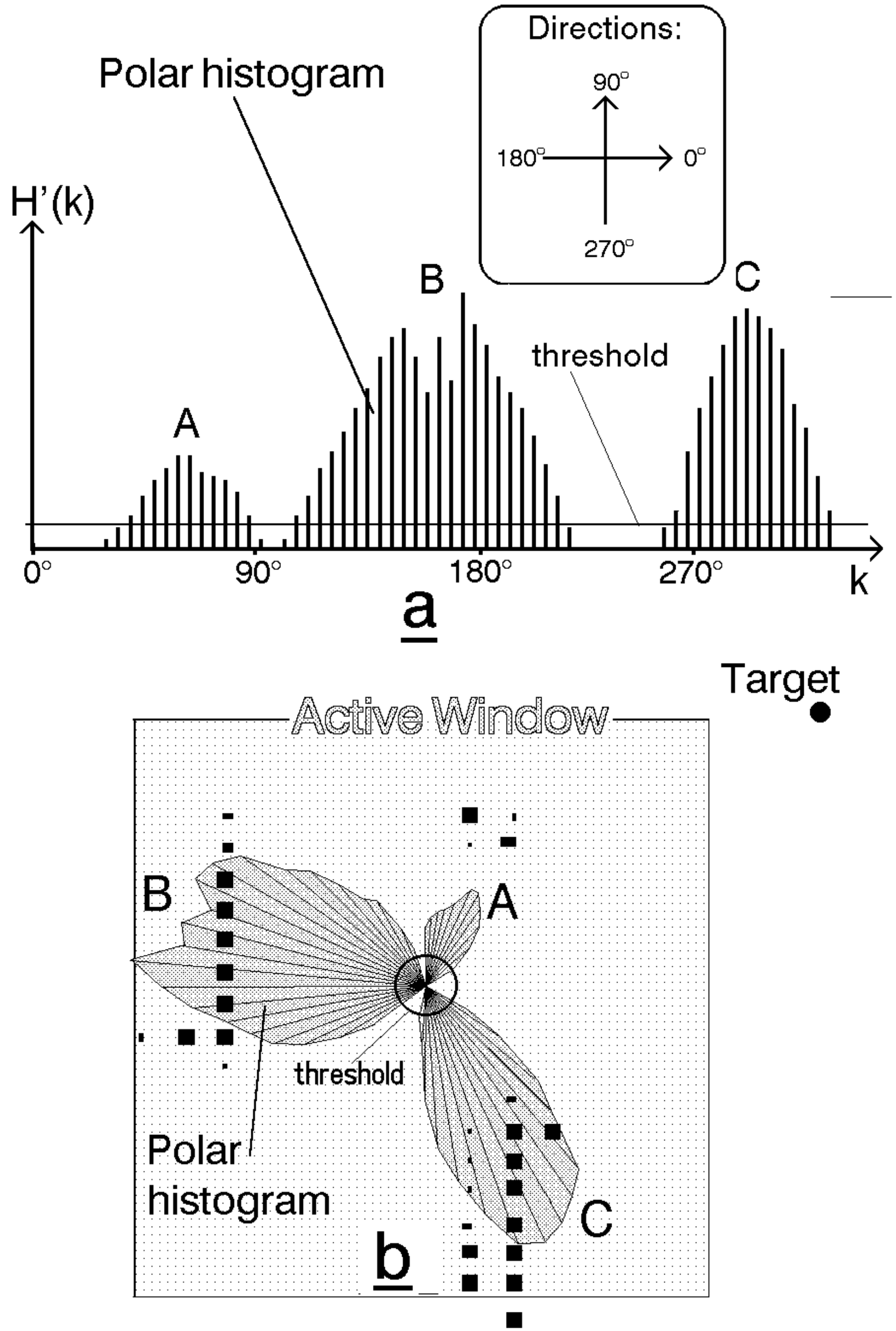
Vector field histogram: Borenstein & Koren

- lab set-up



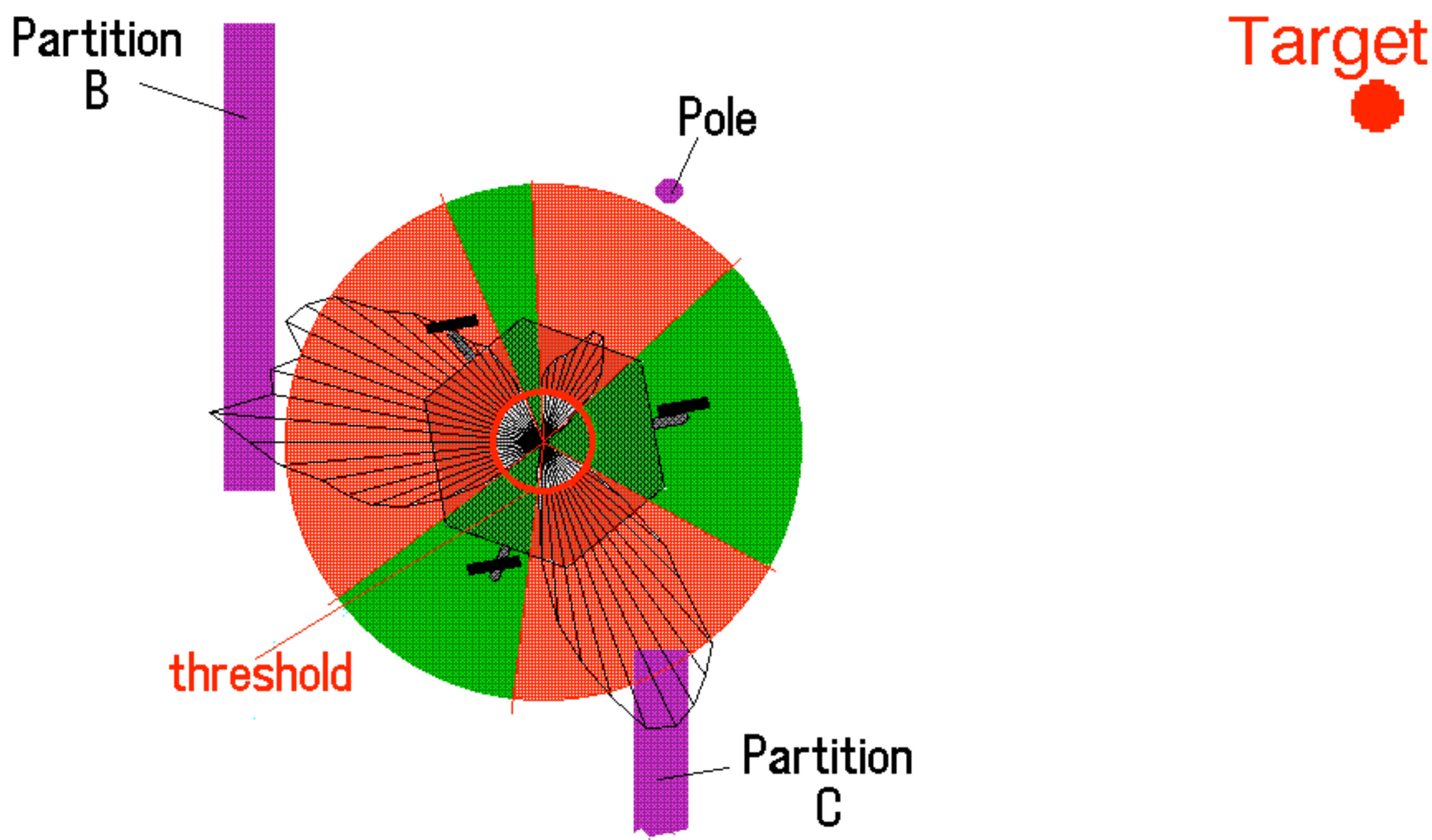
Vector field histogram: Borenstein & Koren

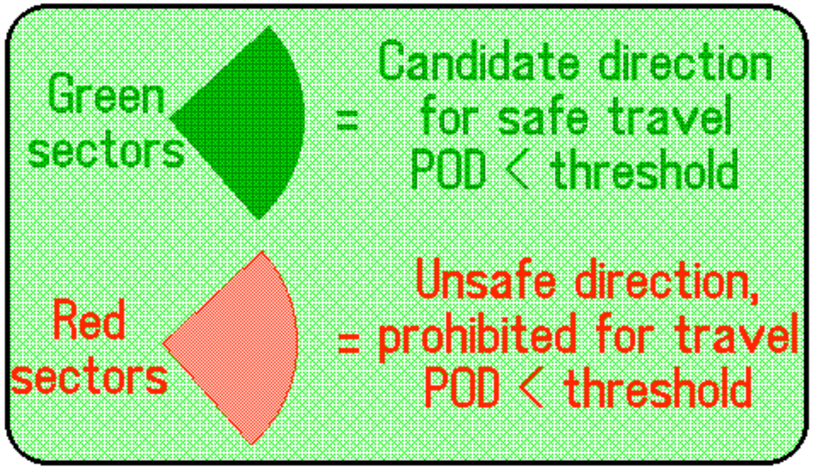
- local polar histogram provides “free” directions



Vector field histogram: Borenstein & Koren

- Select safe direction algorithmically

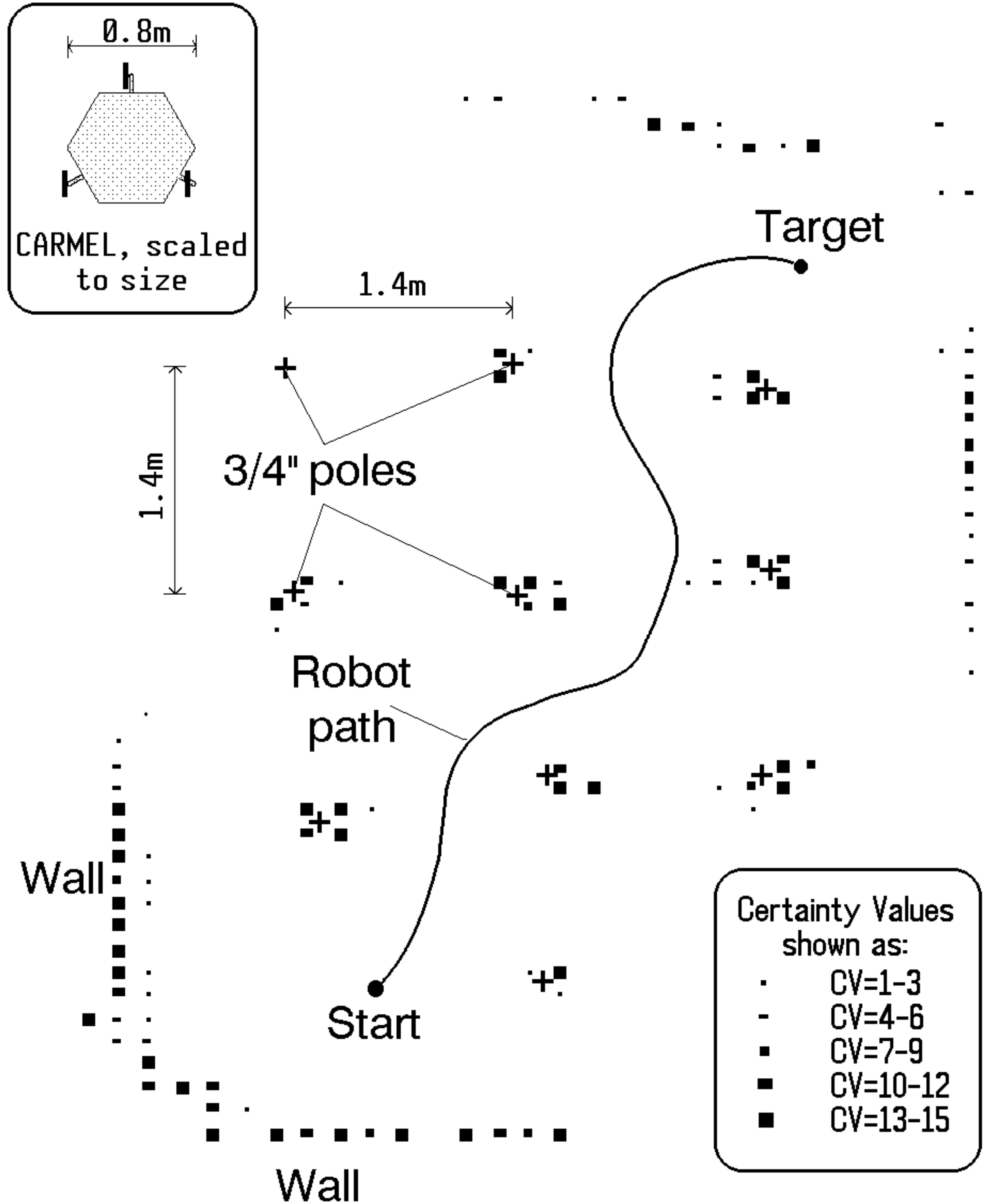




Finding candidate directions for safe travel

Vector field histogram: Borenstein & Koren

- works



Extension of attractor dynamics approach

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

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

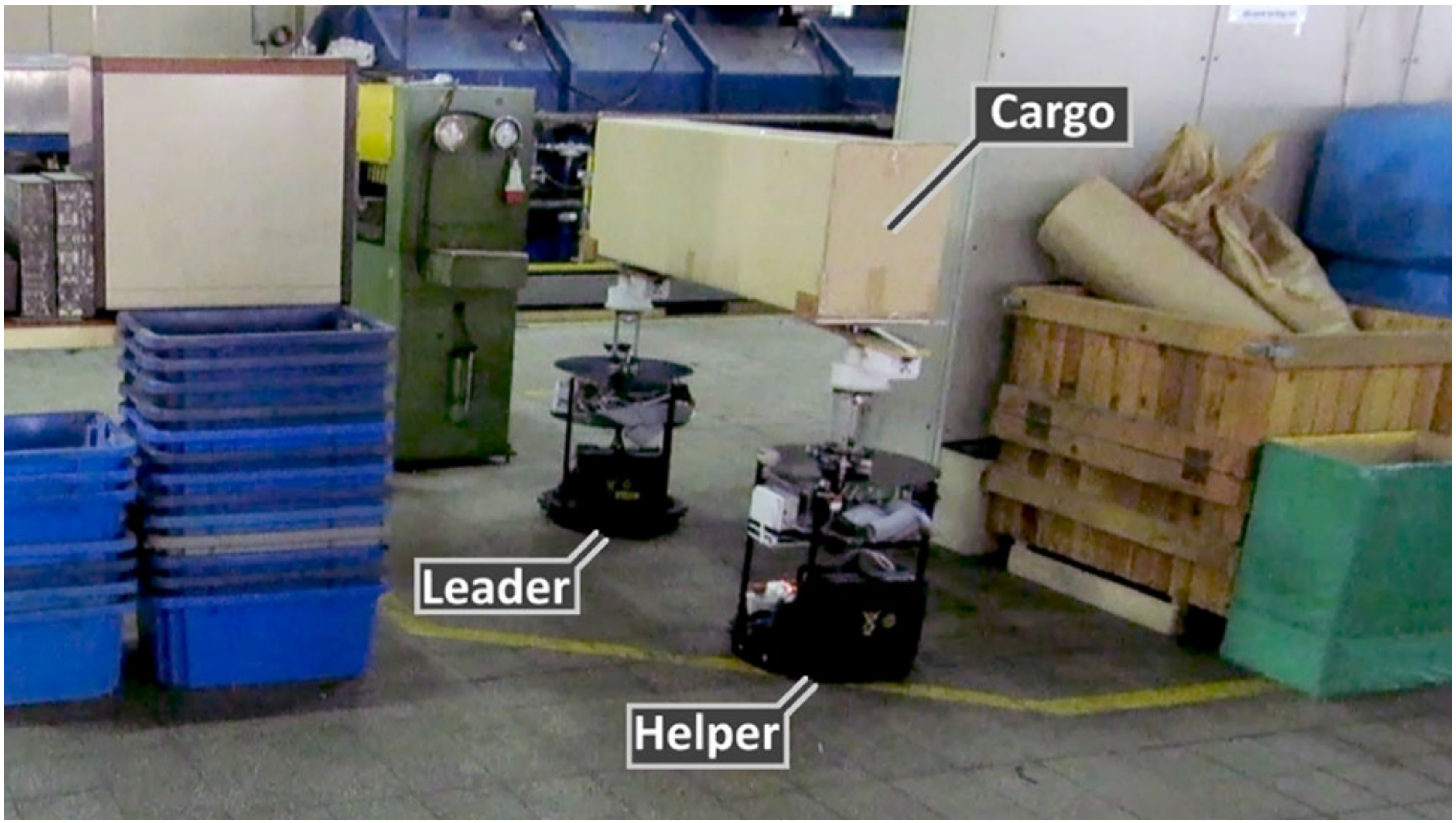
Toni Machado[1] · Tiago Malheiro[1] · Sérgio Monteiro[1] · Wolfram Erlhagen[2] · Estela Bicho[1]

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

Abstract

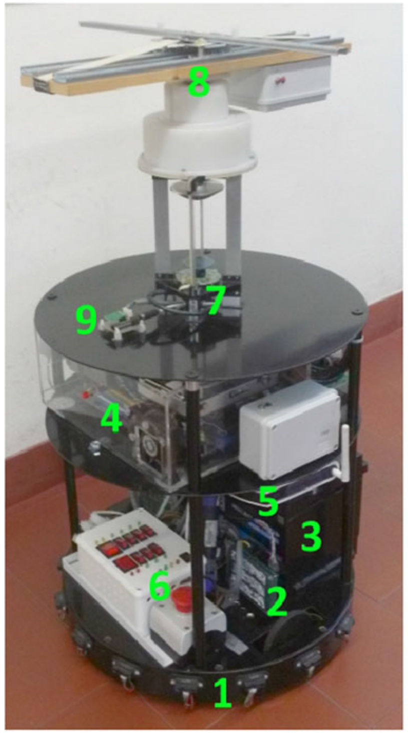
This paper shows how *non-linear attractor dynamics* can be used to control teams of two autonomous mobile robots that coordinate their motion in order to transport large payloads in unknown environments, which might change over time and may include narrow passages, corners and sharp U-turns. Each robot 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



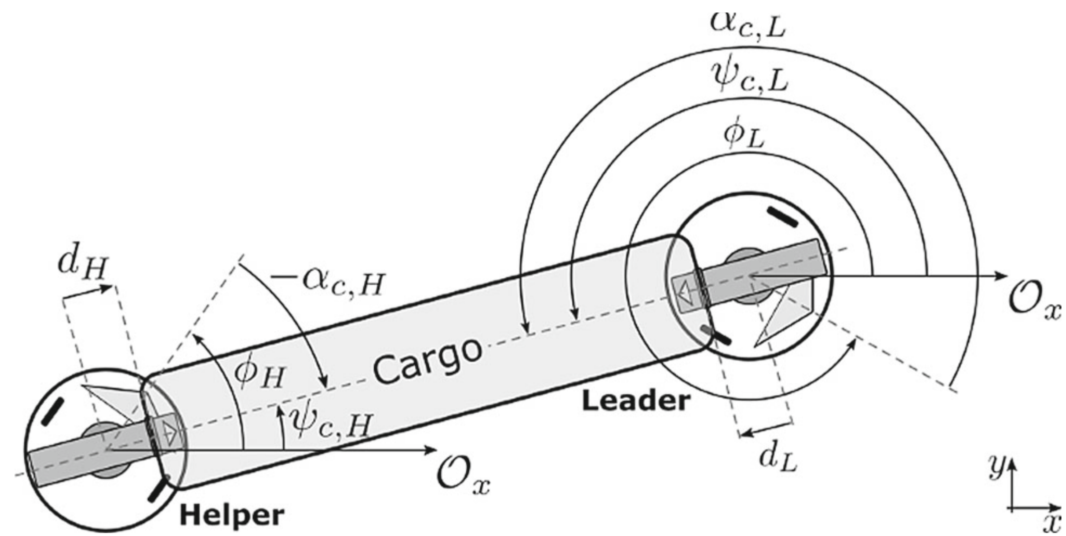
[Machado et al, 2019]

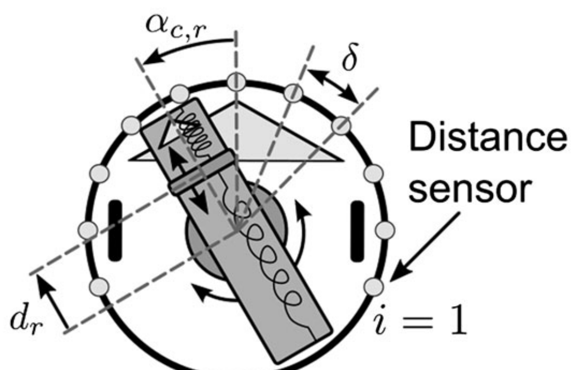
Extension of attractor dynamics approach



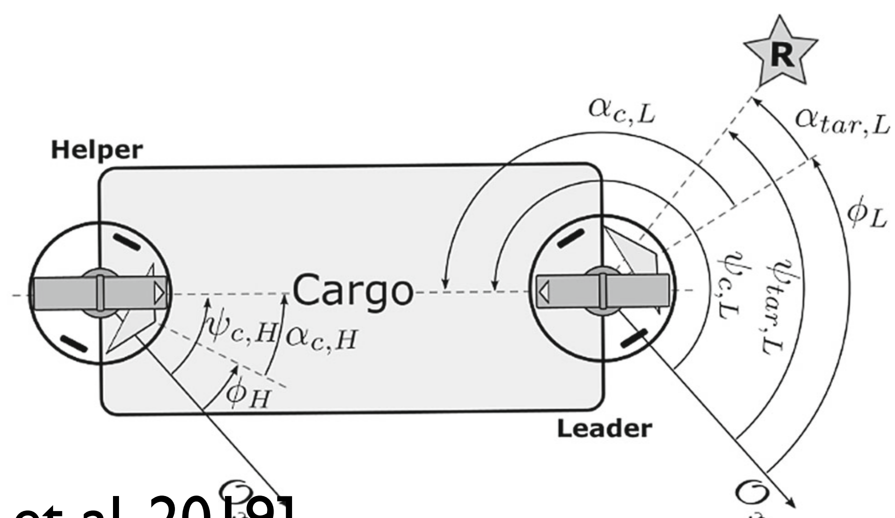
1. Distance sensor
2. Motorized wheel
3. Battery
4. Computer
5. Wireless router
6. Power adapter
7. Vision System
8. Support 2 DoF
9. Compass

(a)



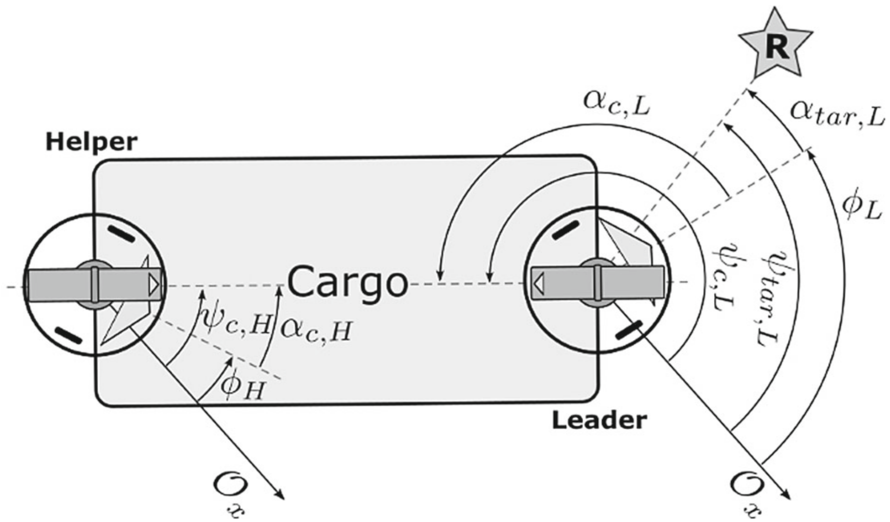


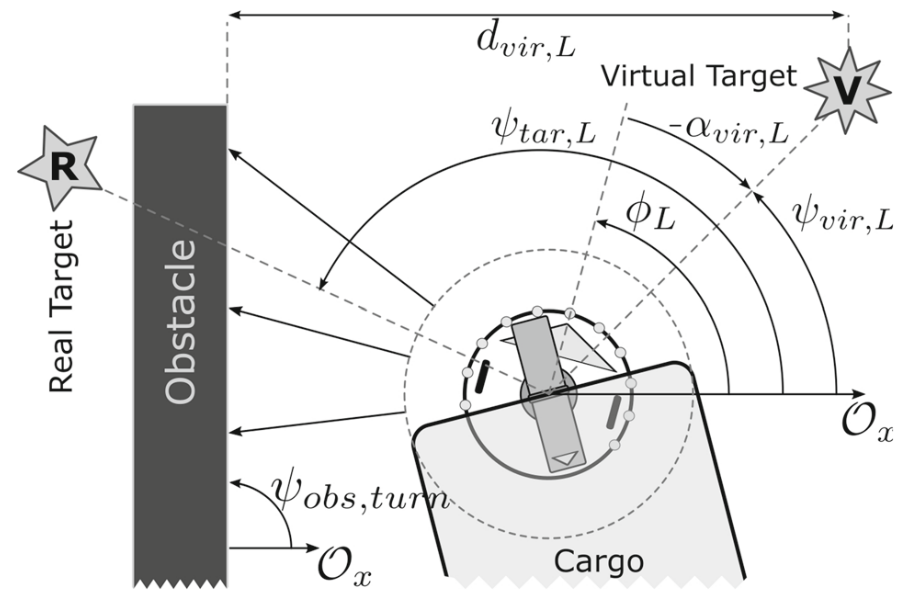
(b)

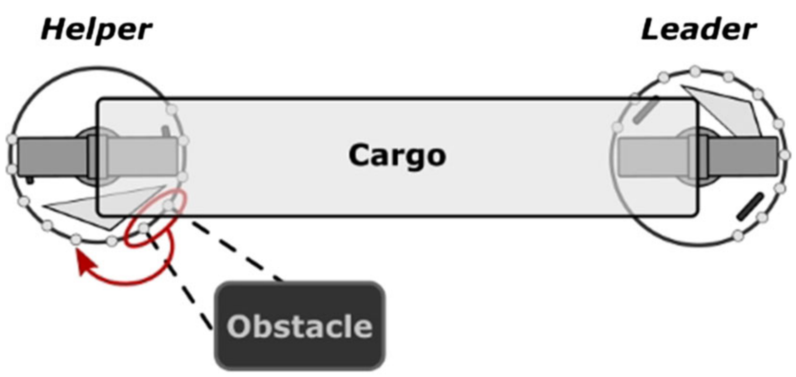


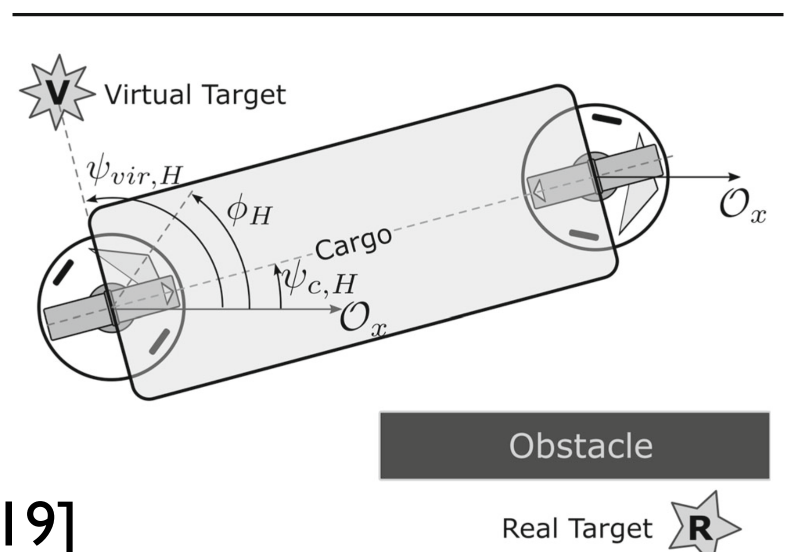
[Machado et al, 2019]

Extension of attractor dynamics approach



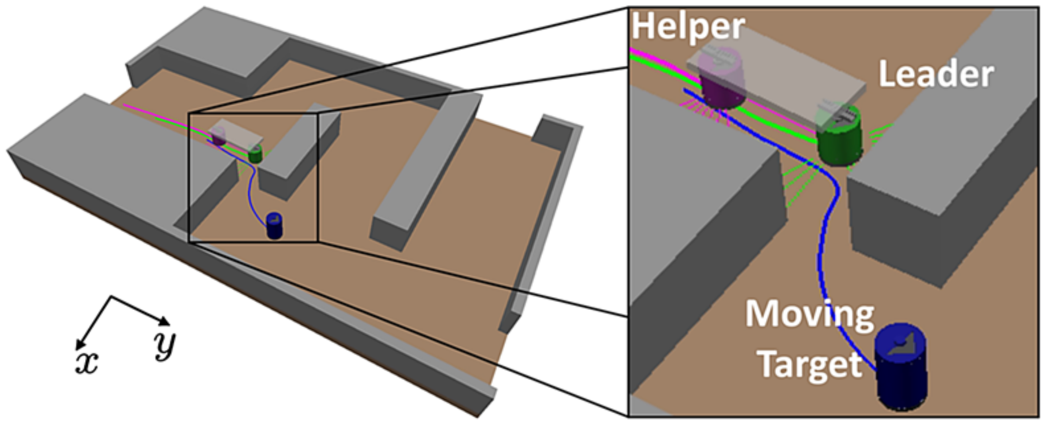




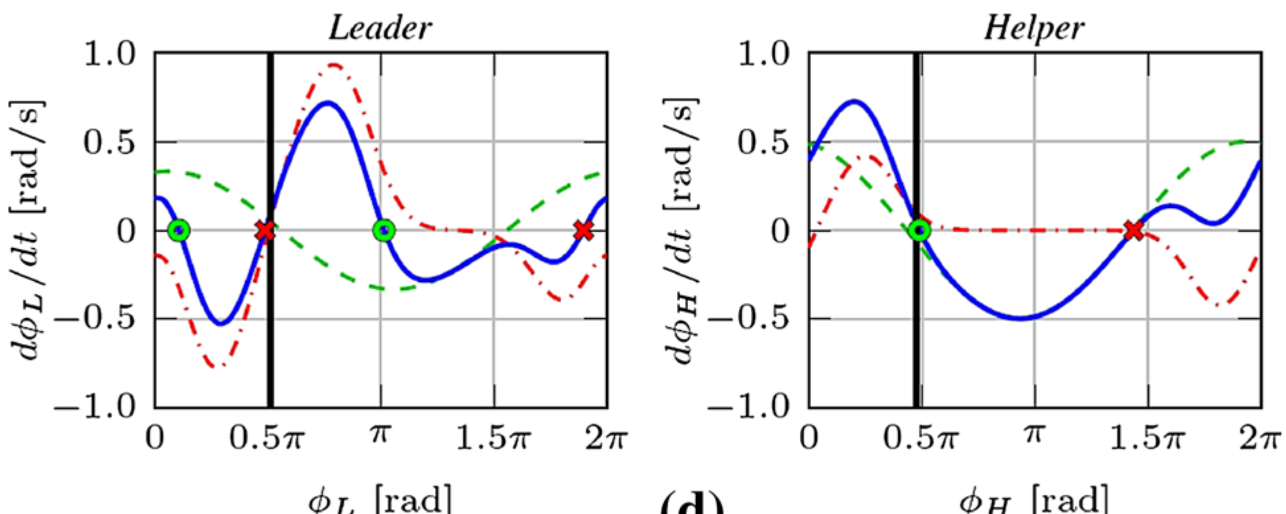


[Machado et al, 2019]

Extension of attractor dynamics approach



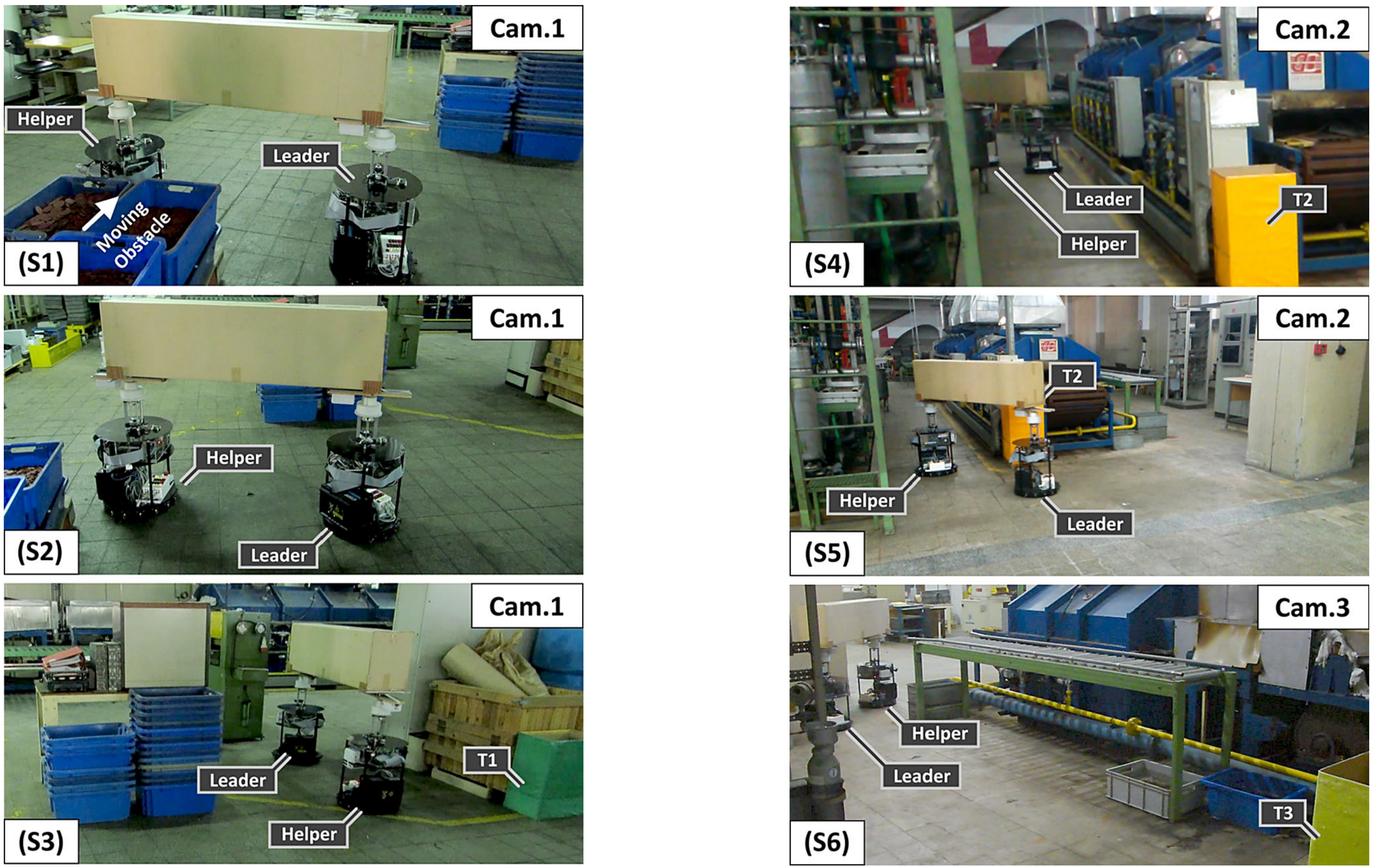
(c)



(d)

[Machado et al, 2019]

Extension of attractor dynamics approach



[Machado et al, 2019]

Summary

- powerful approaches exist for motion planning, which is computational hard in theory and practice
- 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