LECTURE 13

Schema theory

Reactive paradigm

Ola Ringdahl

UMEÅ UNIVERSITY

CONTENTS OF THIS LECTURE

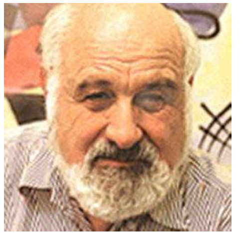
- Biological Foundations of Robot Behavior (part 2)
  - Schema theory
- Reactive paradigm
  - Subsumption
  - Potential fields

SCHEMA THEORY

UMEÅ UNIVERSITY

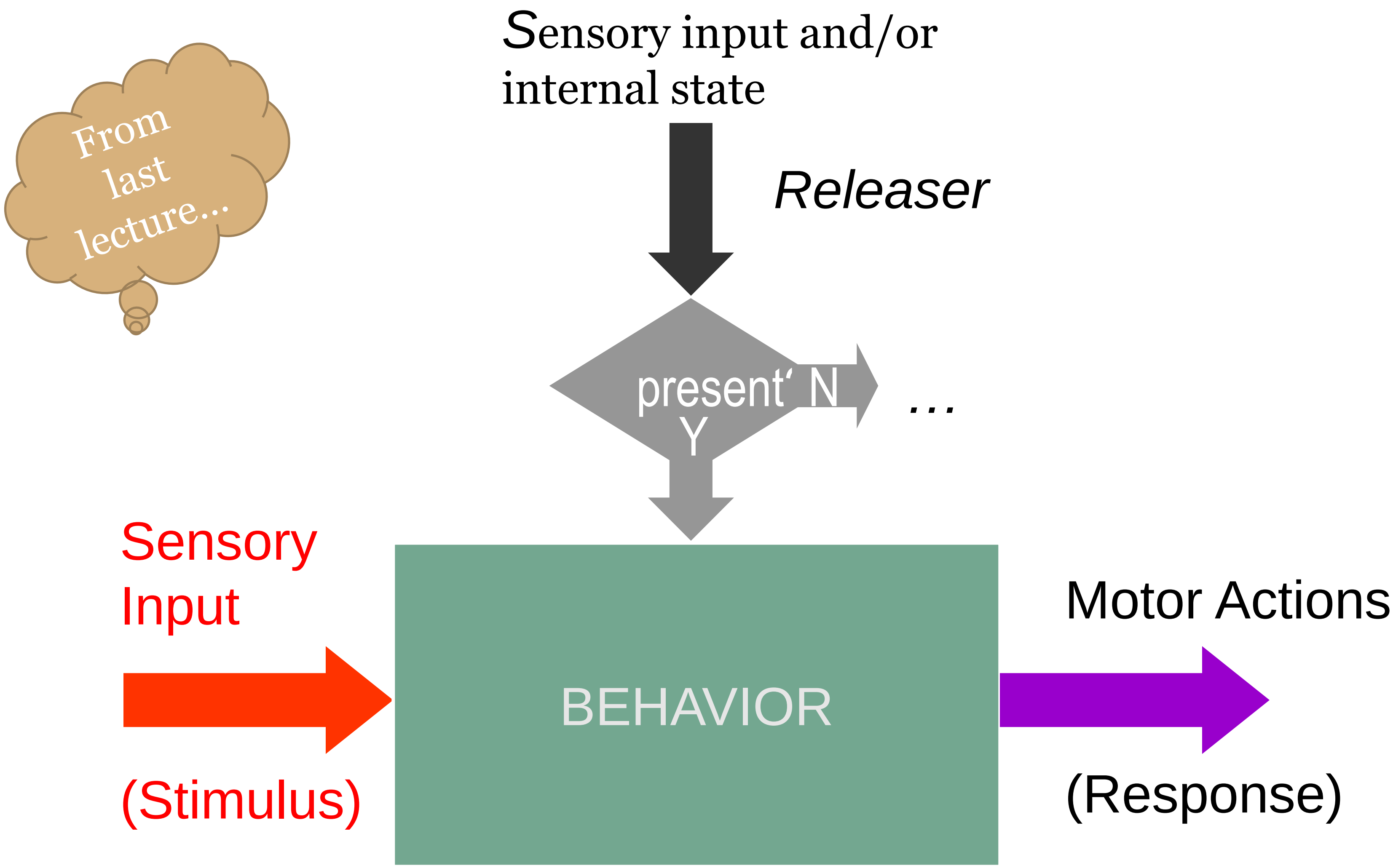
SCHEMA THEORY

- A way to describe and model behaviours
- Inspired by psychology (used there since the early 1900's)
- Applied to robotics by Michael Arbib



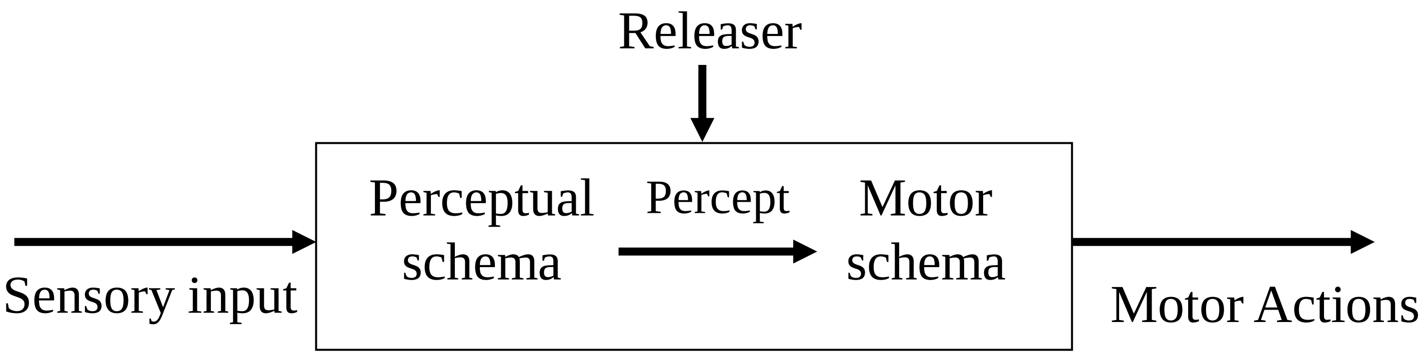
- A schema is a generic template for performing an activity. It contains (compare with OOP):
  - Models and algorithms for how to act and/or perceive
  - Data structures (not in purely reactive behaviours)

INNATE RELEASING MECHANISMS (IRM)



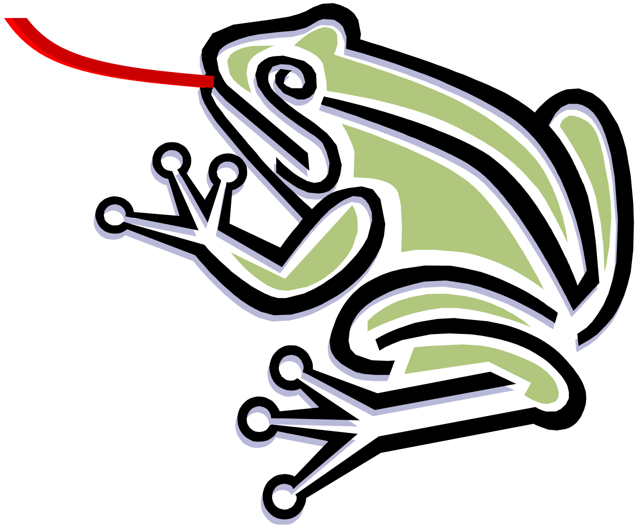
BEHAVIORS AS SCHEMAS

- The Perceptual Schema produces a Percept
- The Percept is read by the Motor Schema
- Motor actions are vectors: direction and strength

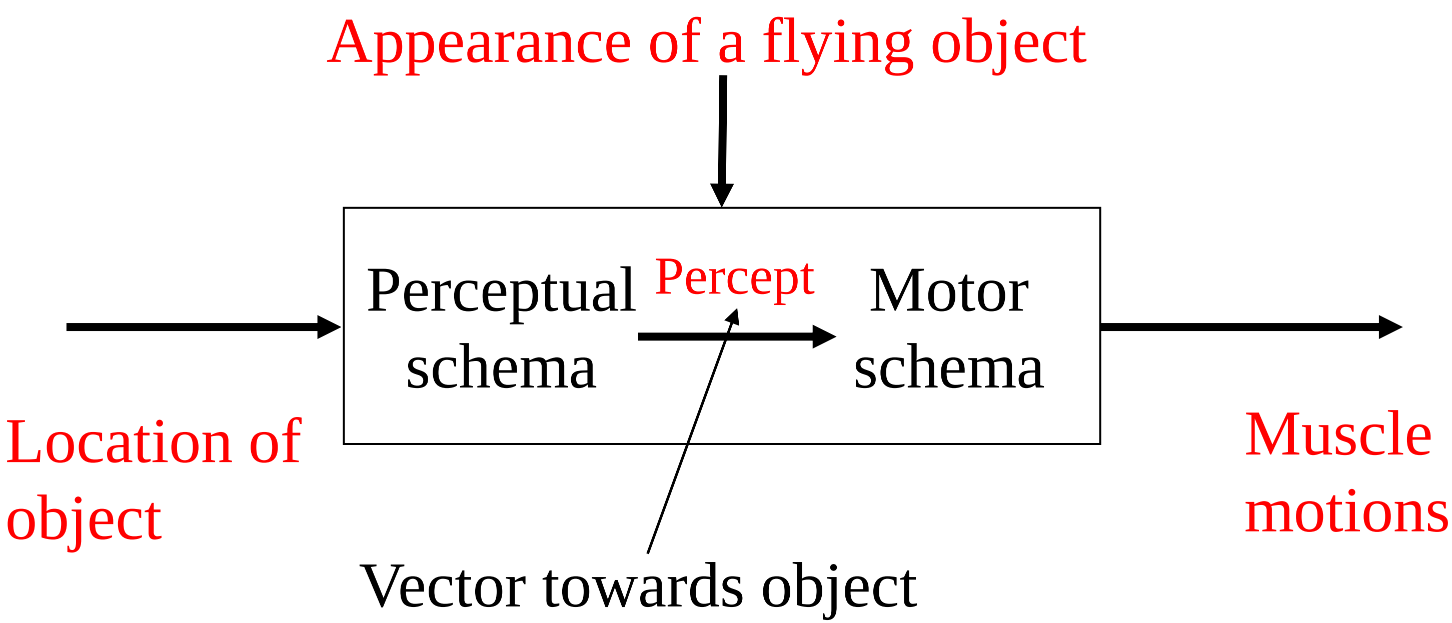


- Several schemas may be active at the same time
- Actions are combined by vector addition

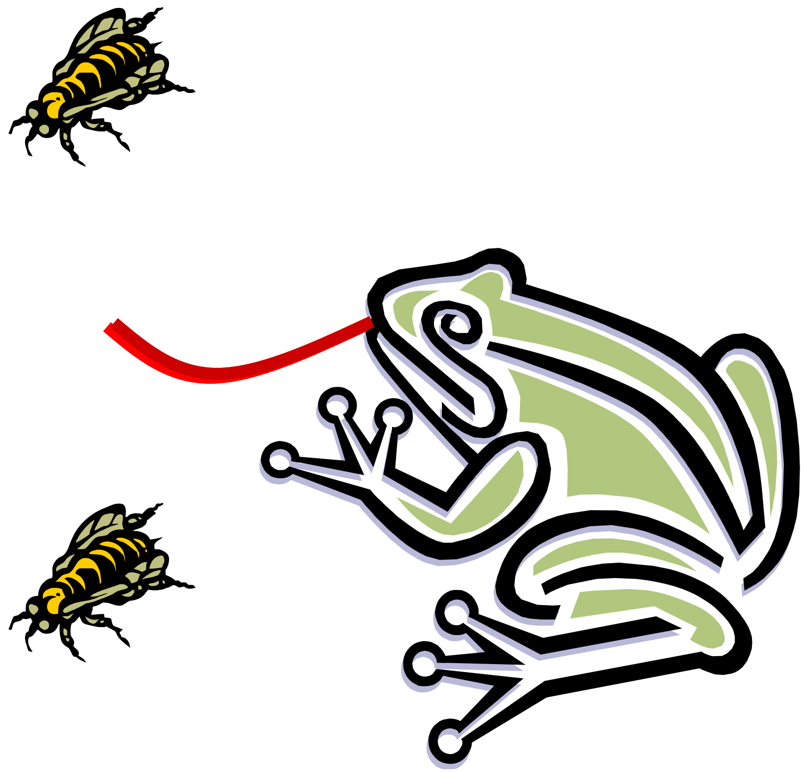
Fly Snapping Behavior



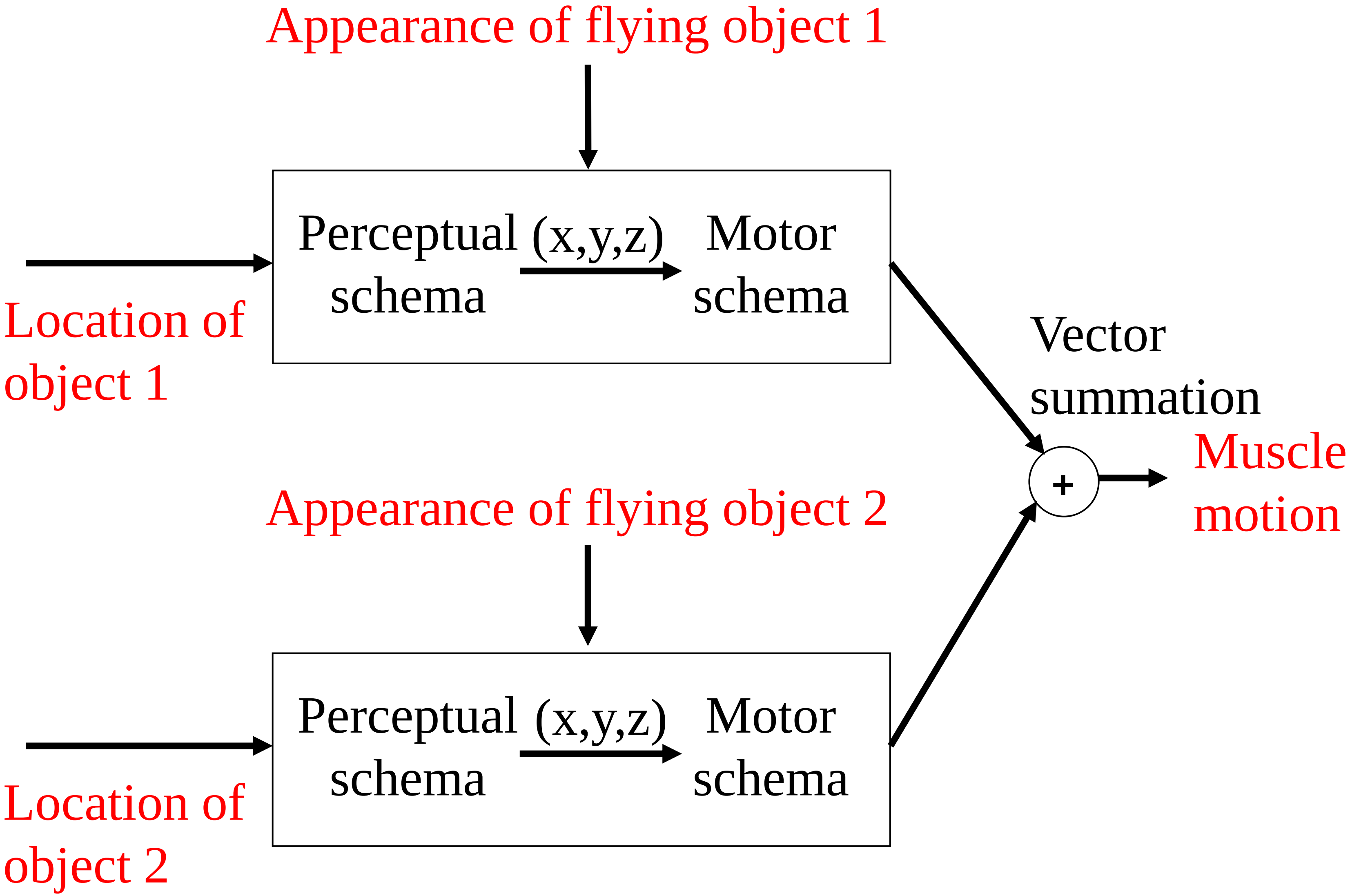
Fly Snapping Behavior



2 flies at the same time



2 flies at the same time



The toad will indeed snap in the middle!

1 fly may be more than enough

CONCLUSIONS FOR ROBOTICS

- Maybe programming in terms of parallel behaviors is better than STRIPS or other deliberative approaches
- Individual behaviors
  - are often simple
  - are independent
  - can be implemented as threads
- Sequences of behaviors
  - can appear from parallel behaviors activated by releasers (internal states and/or external sensor data)
  - may appear intelligent for an observer
- The system "degrades gracefully" if any part fails since it's modular
- IRMs and Schemas are nice ways to start thinking about the computational structure of programming a robot

REACTIVE PARADIGM

Mataric - Reactive paradigm (p163-169)

THE REACTIVE PARADIGM

- Reacted against classical AI
  (the hierarchical paradigm/ deliberative systems)
- Reduced emphasis on:
  - Knowledge representation
  - Planning
- Increased emphasis on:
  - Sensing and acting within the environment
  - ”Situatedness” - interaction with the environment is crucial for intelligent behavior
  - ”Embodiment” – the structure of the body is tightly connected to behavior and intelligence.

THE REACTIVE PARADIGM

Suggested by Rodney Brooks MIT (1986)

- “Planning is just a way of avoiding figuring out what to do next”
- “The world is its own model”
- “Elephants don’t play chess”
- Complex control systems are not necessary for complex behavior
- Intelligence is in the eye of the observer
- No representation. No calibration. No complex computers. No High bandwidth communication

RELATED QUOTES

”Everybody’s got a plan until they get hit”

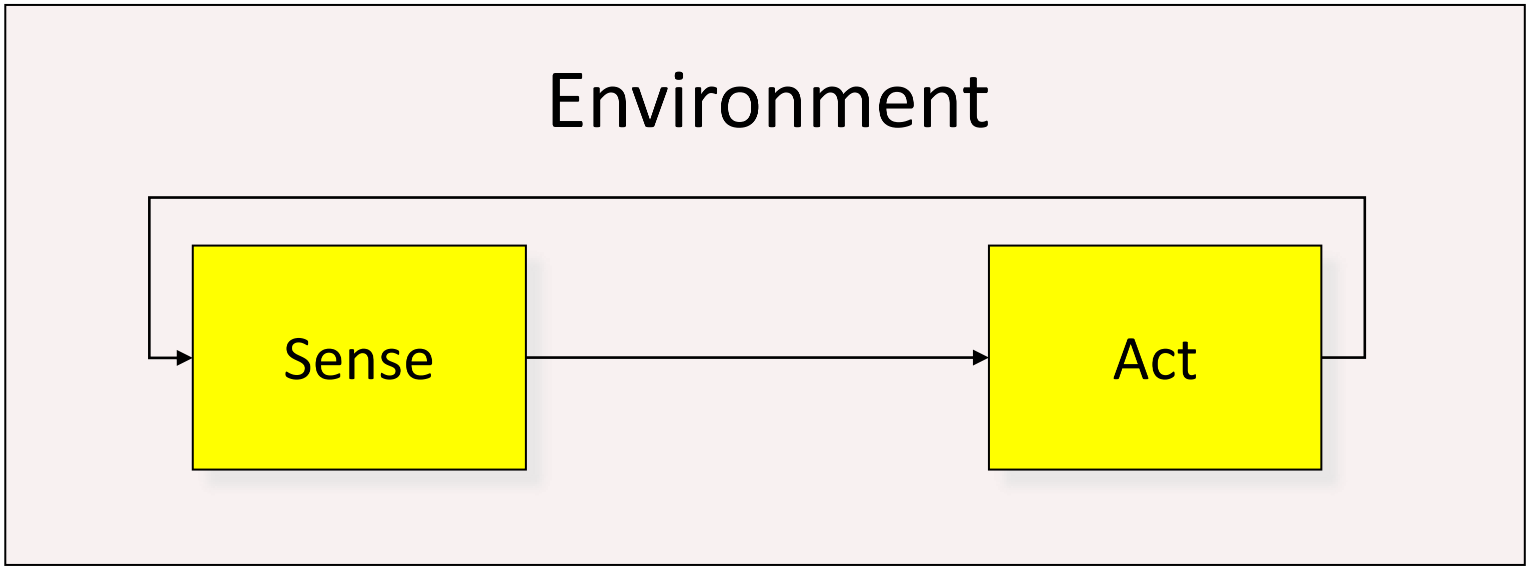
Mike Tyson – unknowingly commenting on
The closed world assumption

"If you want to make God laugh, tell him about your plans”

Woody Allen – general disbelief in predictions of the future

THE REACTIVE PARADIGM

Walter 1953, Breitenberg 1984, Brooks 1986



Method: “Sense-Act

- Sensor data is immediately transformed into Actions
- Based on biology (e.g. insects)
  - Stimuli → Response

THE REACTIVE PARADIGM

- The basic components are called *BEHAVIORS*
- Each behavior
  - is implemented as reactive rules, mapping stimuli to respones without thinking or computing
  - has no (or little) memory
  - computes responses continously
- All behaviors exist in parallell and are activated (triggered) by percepts from the sensors
- *Coordination* if several behaviors are active at the same time

ARCHITECTURAL ISSUES

- How to represent behaviors
- How to trigger behaviors
- How to coordinate behaviors

<u>Two common architectures:</u>

- Subsumption architecture
- Potential fields

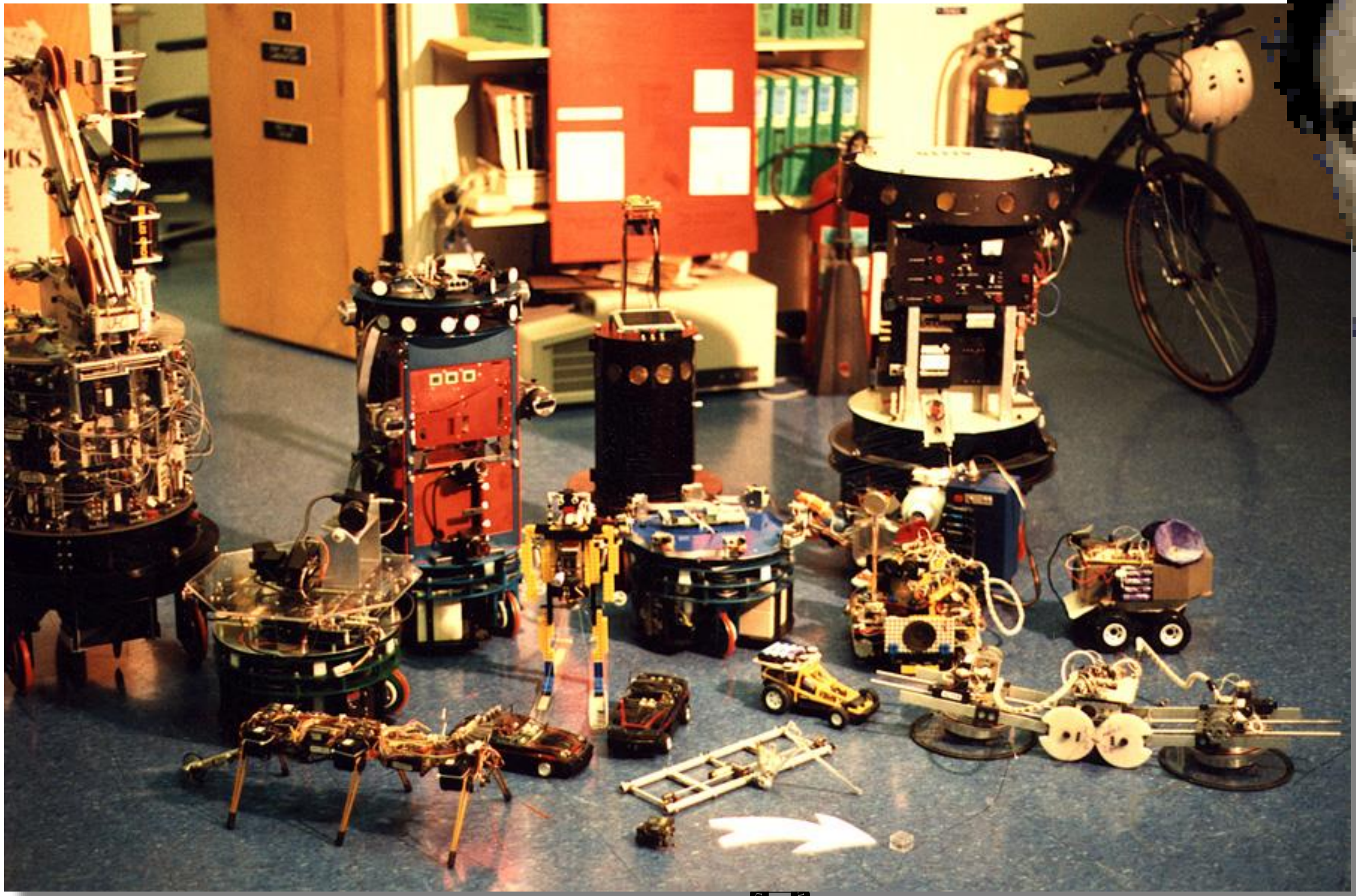
REACTIVE PARADIGM ALGORITHMS

Subsumption

Potential fields

Subsumption Architecture

Suggested by Rodney Brooks MIT (1986)

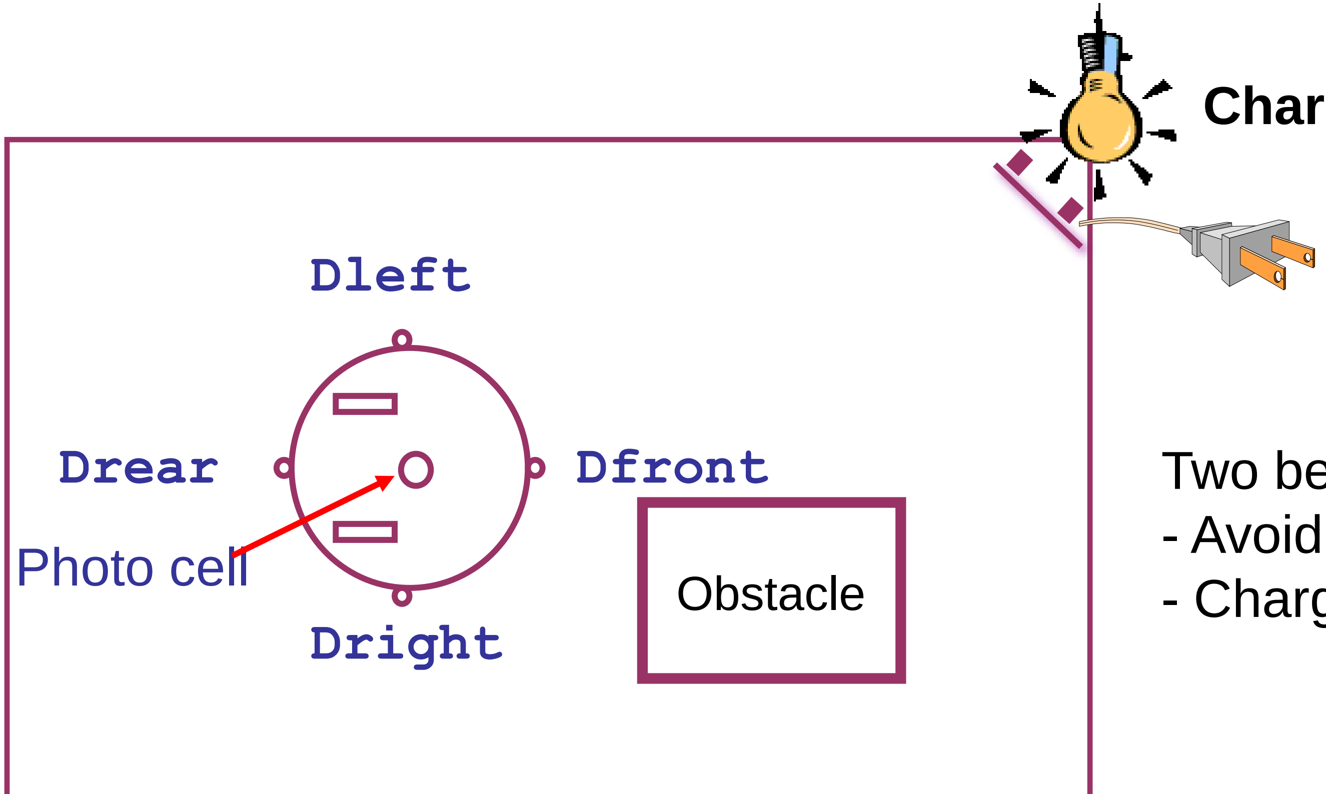


Example

Subsumption Architecture

A self-charging robot

- 2 engines for propulsion and stearing
- Controlled by speed and angular velocity
- Battery voltage sensors `Bat`
- 4 ultrasonic distance sensors `Dleft,Dright,Dfront,Drear`
- 1 photocell that gives the angle `D` to the light source



Two behaviors:
- Avoid
- Charge

Example

Subsumption Architecture

Avoid

US sensor → *Avoid* → Motors

The behavior *Avoid* contains code that maps the signals from the 4 US sensors to the actuator (Motors):

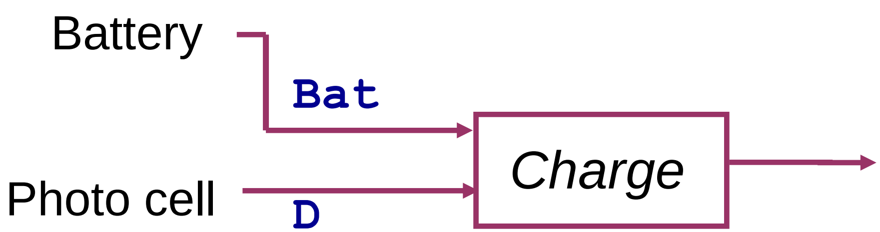
```
If Dfront<d0 or Drear<d0 or Dleft<d0 or Dright<d0
   % Act only if an obstacle is close
   if Drear<min(Dleft, Dright, Dfront) and Dfront>S0
     "move ahead"   % Go away from obstacles
   else
      if Dleft<Dright then "turn right"
      if Dright<Dleft then "turn left"
   end
end
```

Note: responses are computed continously (don't wait until finished)

Example

Subsumption Architecture

Charge



The behavior *Charge* contains code that turn the robot towards a light source (the charger), move forward and dock.

The PhotoCell senses light in direction **D** (**D** = 0 if light is straight forward)

For instance:

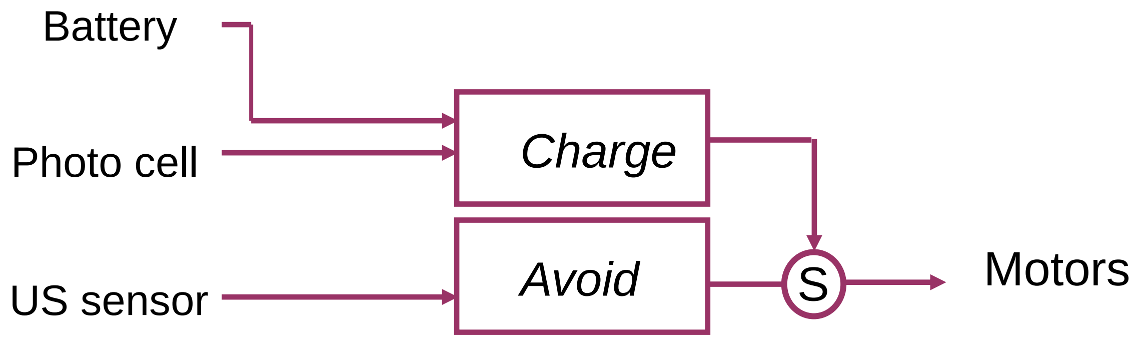
```
If Bat<b0 and
   if D>Dtol then "turn right"
   if D>-Dtol then "turn left"
   "move ahead"
End
```

Note: responses are computed continously (don't wait until finished)

Example

Subsumption Architecture

Coordination of Behaviors

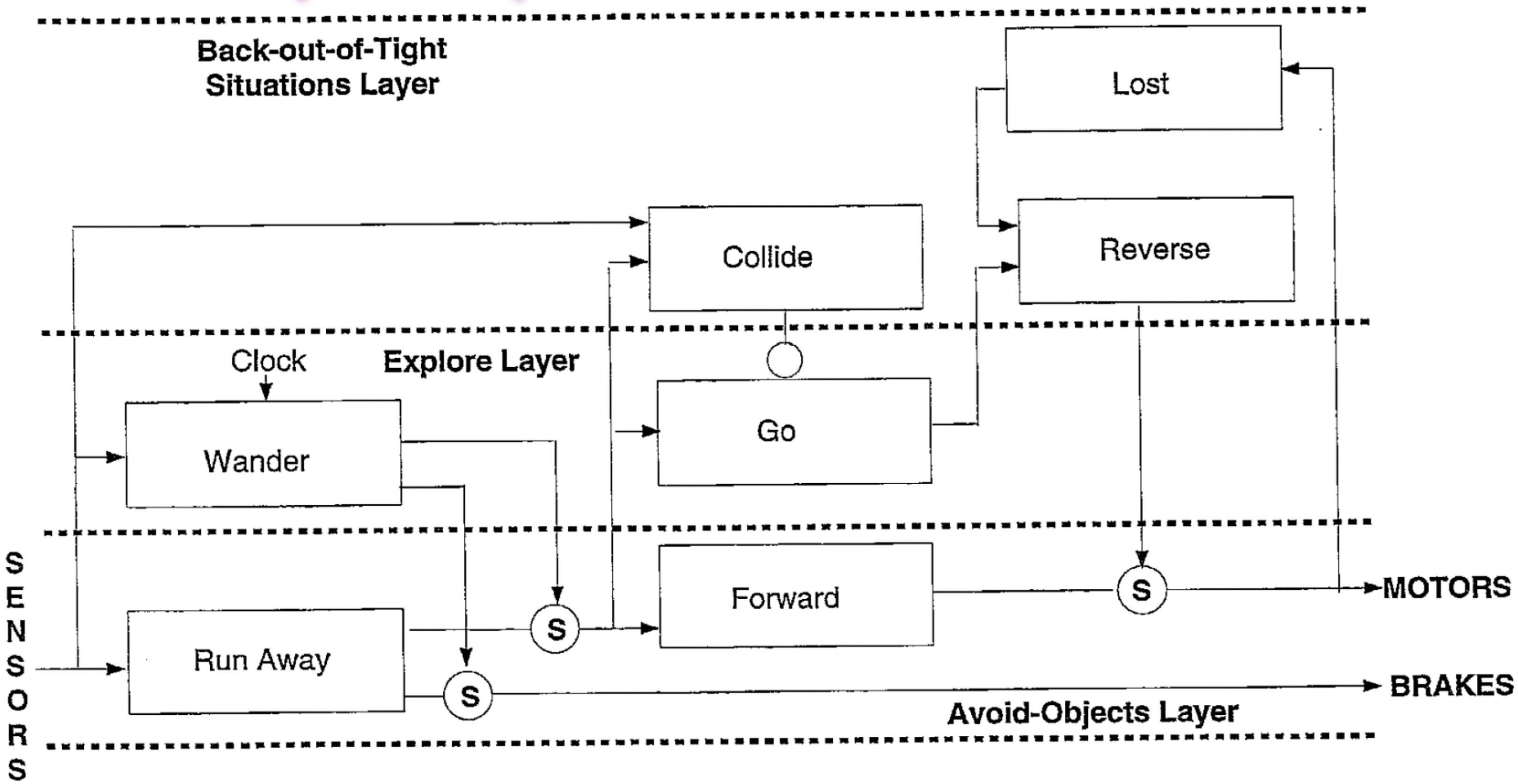


- S is a *suppressor node* that blocks and replaces messages from the *Avoid* behavior iff *Charge* sends messages at the same time.
- *Charge* subsumes (contains/*innesluter*) *Avoid*
- *Avoid* is subsumed by *Charge*

UMEÅ UNIVERSITY

Subsumption Architecture

A more complex example



Each layer has a separate goal

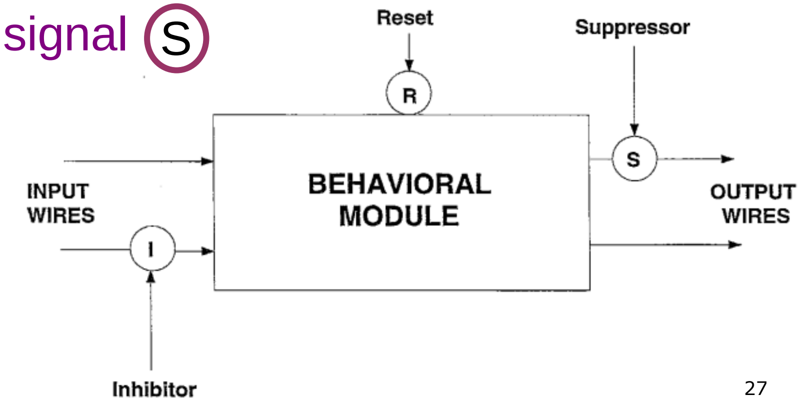
UMEÅ UNIVERSITY

Subsumption Architecture

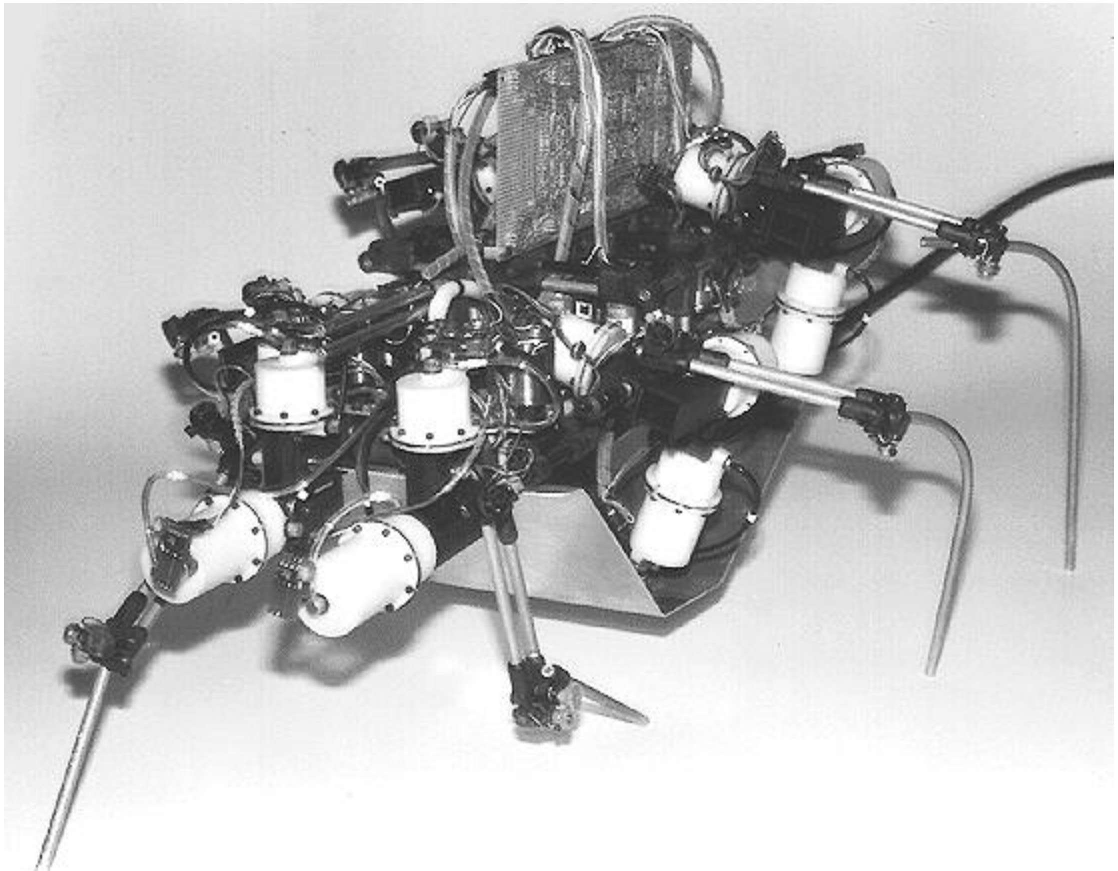
- Each task-achieving layer contains one or several reactive behaviors
- (almost) no internal states
- Loosely coupled layers:
  - The lower layers have no awareness of higher levels (i.e.: bottom-up design)
  - Higher layers may access lower levels, but very low bandwidth. “The world serves as primary medium of communication”
- Competitive coordination with two mechanisms:

  - Inhibition. Prevents a signal from being transmitted I

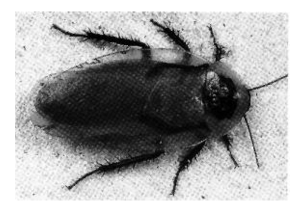
  - Suppression. Blocks and replaces a signal S

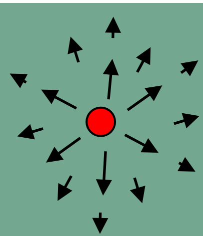


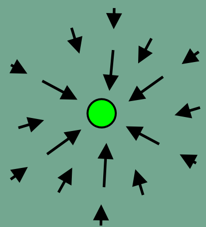
A robot cockroach



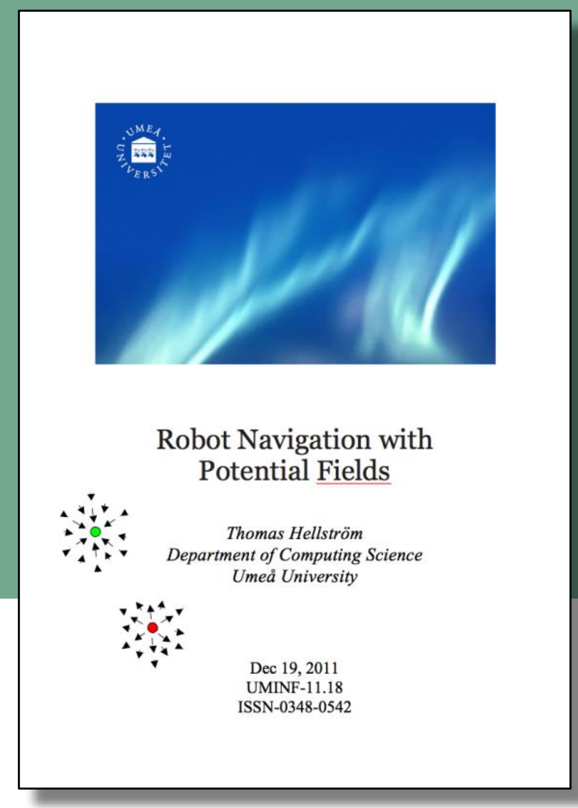
- A six-legged, pneumatically powered walking robot based on the Subsumption architecture with 57 behaviors
- The robot's mechanics are modeled after a cockroach, Blaberus Discoidalis.







POTENTIAL FIELDS FOR ROBOT NAVIGATION


Robot Navigation with Potential Fields

Thomas Hellström
Department of Computing Science
Umeå University

Dec 19, 2011
UMINF-11.18
ISSN-0348-0542

UMEÅ UNIVERSITY

POTENTIAL FIELDS

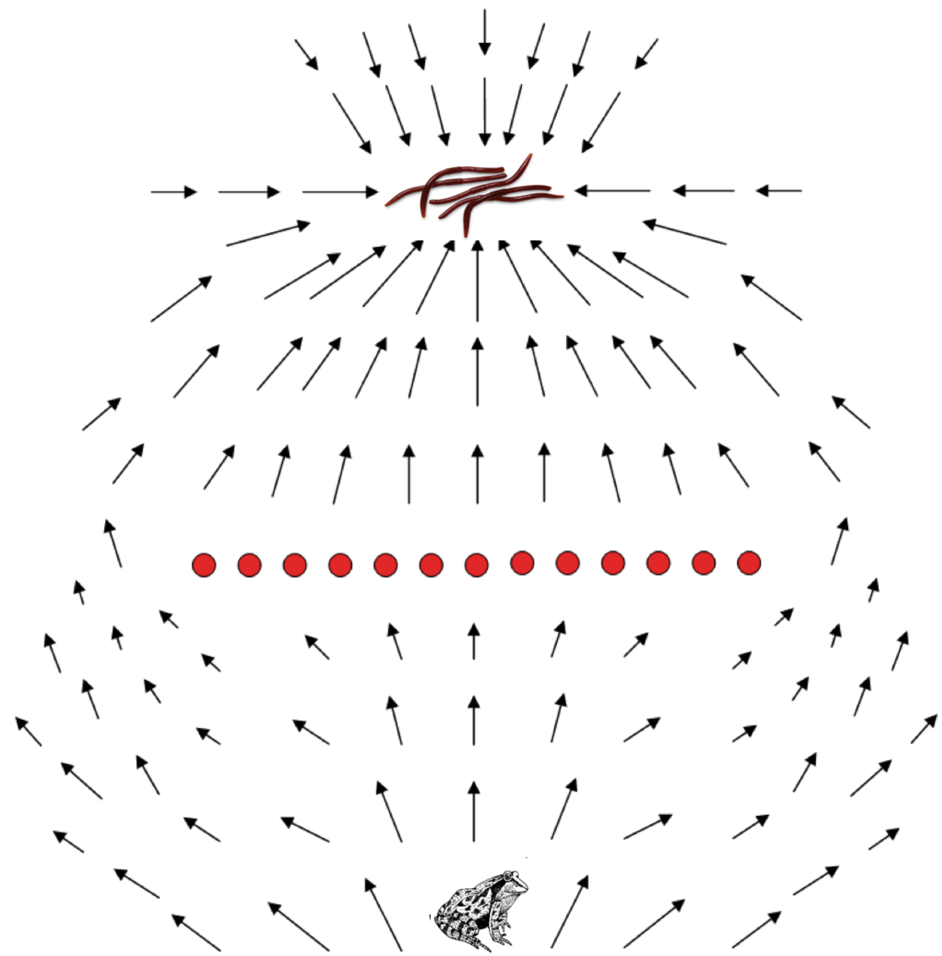
- A reactive technique for guiding a robot from A to B
  - No Planning!
- Inspired by biology
- Responses are represented in a uniform format:
  Vectors with Strength (force or velocity) and Orientation
- Can be used to implement Motor schemas

INSPIRED BY BIOLOGY

Describe the “forces” emanating from the shark and men below

INSPIRED BY BIOLOGY

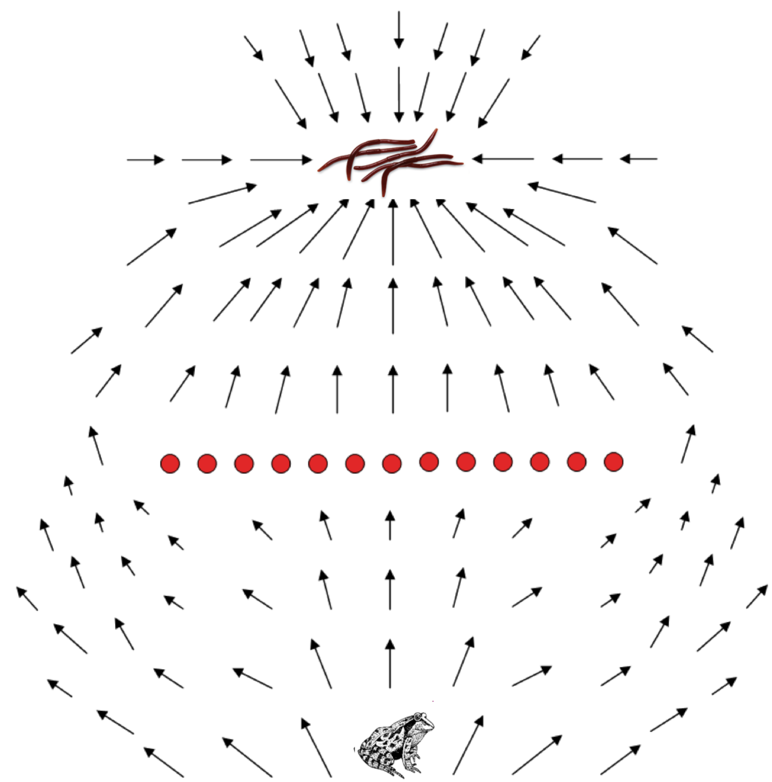
- Motion of toads attracted to a can of worms
- Model: Attraction and repulsion



UMEÅ UNIVERSITY

THEORY

- A robot should navigate over a 2D field occupied by an obstacle, until it reaches a goal position G.
- For each point (x,y) we define two values given by *potential field functions*, e.g.:
    - U_{rep}: 1/(distance to obstacle)
    - U_{att}: distance to G
- We want to move in the direction of decreasing U_{rep} and U_{att}



THEORY

- For a function f, the gradient vector ∇f points in the direction of increasing f.
- We define *force vectors F* as the negative gradients of U_{rep} and U_{att}:

$$F_{rep} = -\nabla U_{rep} = -\left(\frac{\partial U_{rep}}{\partial x}, \frac{\partial U_{rep}}{\partial y} \right)$$

$$F_{att} = -\nabla U_{att} = -\left(\frac{\partial U_{att}}{\partial x}, \frac{\partial U_{att}}{\partial y} \right)$$

- F_{rep} points in the directions of smaller U_{rep} (away from the obstacle)
- F_{att} points in the directions of smaller U_{att} (towards G)
- A strategy to move safely towards G is to move in the direction $F = F_{rep} + F_{att}$

THEORY

Generalization:

- More obstacles
- Other types of potential fields

The general expression for **F** computed at a point (x,y) is:

$$\mathbf{F}(x, y) = \sum_i \mathbf{F}_i(x, y)$$

where $\mathbf{F}_i(x,y)$ is the force vector associated with potential field U_i, i.e. the negative gradient of U_i : $\mathbf{F}_i(x,y) = -\boldsymbol{\nabla} U_i(x,y)$

AN ALGORITM FOR NAVIGATION

An algorithm for robot navigation from a position (x,y):

1. Represent the goal and all obstacles as potential fields.
2. Compute force vectors $\mathbf{F}_i(x,y)$ for all potential fields
3. Sum all force vectors into $\mathbf{F}(x,y)$
4. Move along $\mathbf{F}(x,y)$ with speed proportional to $|\mathbf{F}(x,y)|$
5. Repeat from 2. until goal is reached

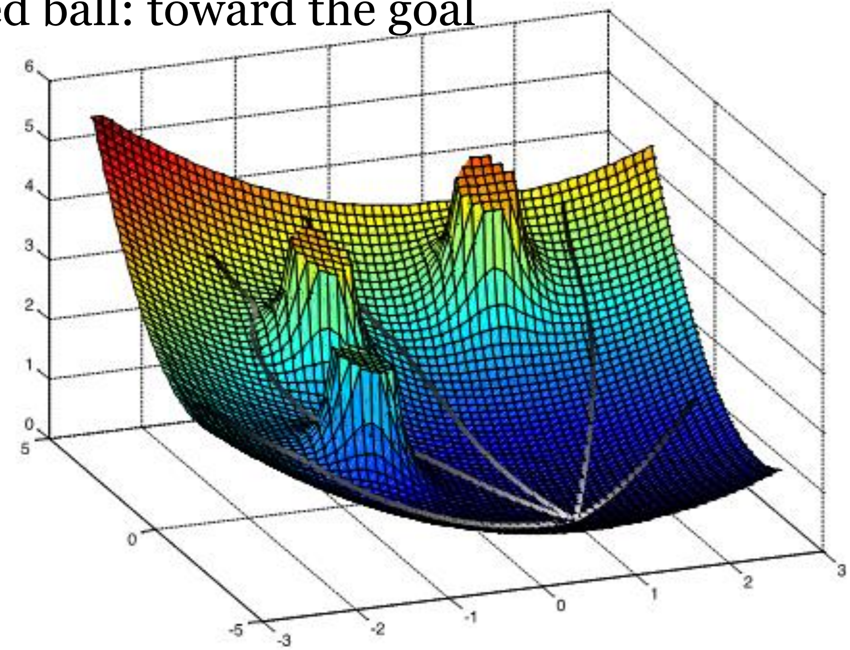
IMPORTANT:

- Normally each $\mathbf{F}_i$ is computed as a function of distance and direction to objects around the robot (from sensors)
- $\mathbf{F}_i$ only has to be computed for current position (x,y)

PHYSICS ANALOGY

A ball finding its way to the bottom of a valley filled with obstacles

- One attractive goal at (1, -2) at the bottom of the valley
- Three repulsive objects as "steep rocks"
- The paths show the motion of the imagined ball: toward the goal avoiding the obstacles



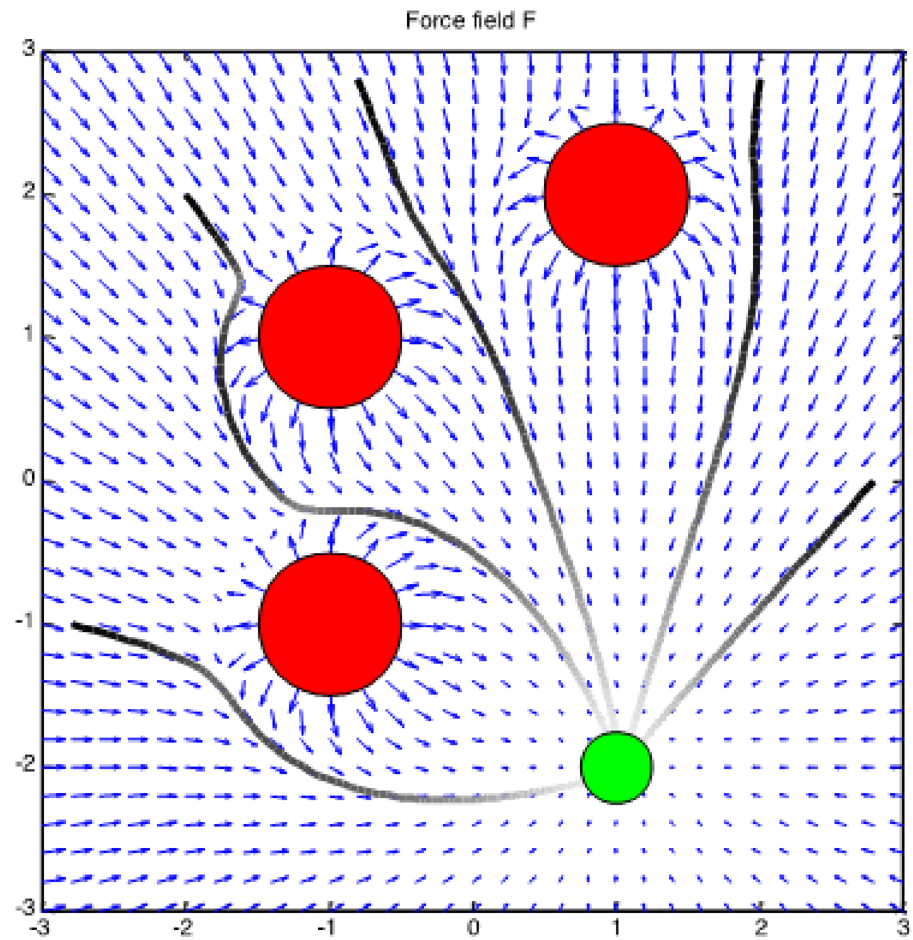
The paths also corresponds to the potential field theory:

- z-axis is the sum U of all potential fields
- Direction of maximum descent $(-\nabla U)$

UMEÅ UNIVERSITY

PHYSICS ANALOGY CONT.

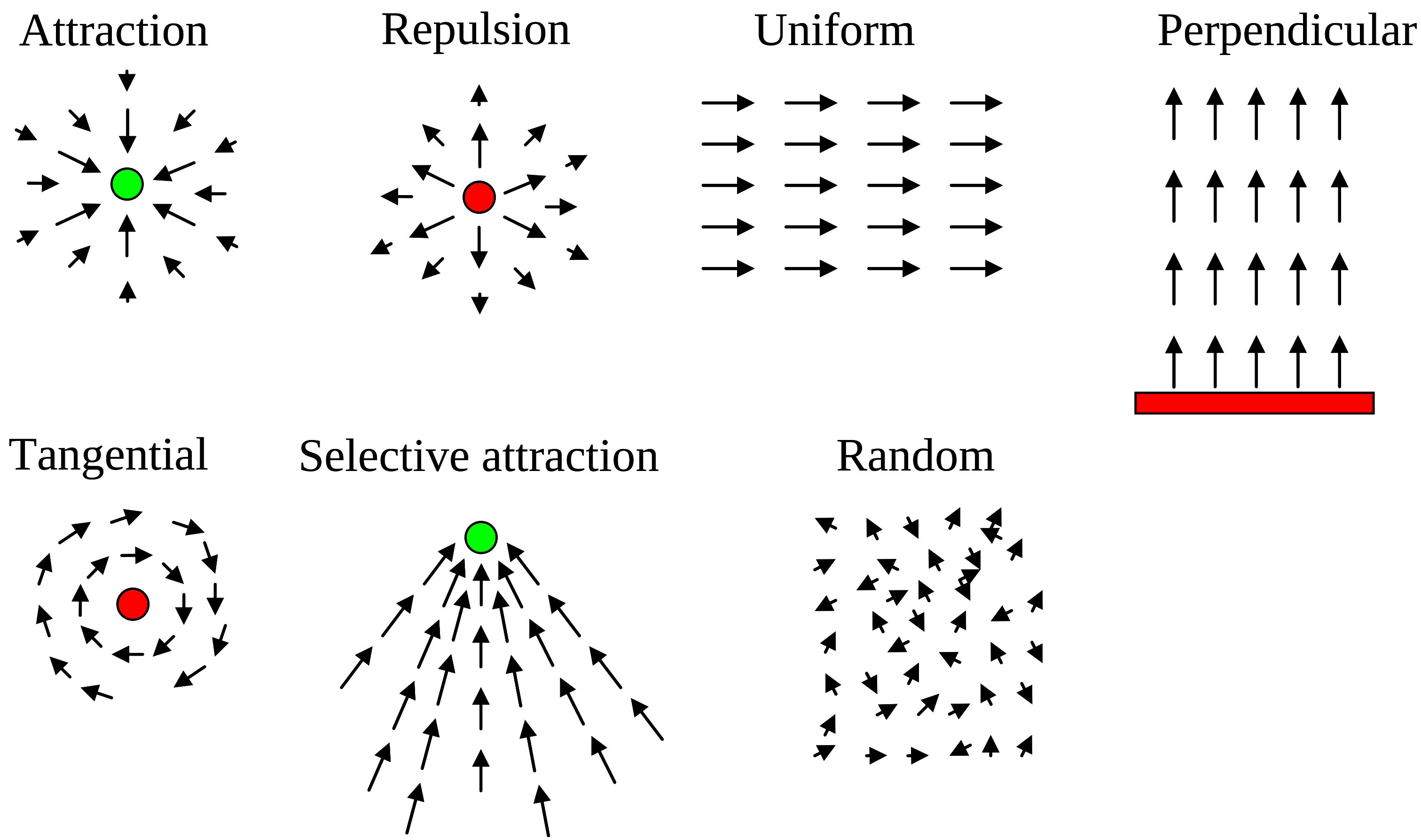
- $\mathbf{F} = -\nabla U$ is the force that causes the ball to move
- Each arrow represents the direction and magnitude of **F** at the location of the arrow
- The robot follows the direction of **F** in every point
- Speed is given by $|\mathbf{F}|$, the magnitude of **F** (shown as gray scale of paths)
- Compare with the toad and worms



Red: repulsive obstacles

Green: attractive goal

BASIC TYPES OF POTENTIAL FIELDS



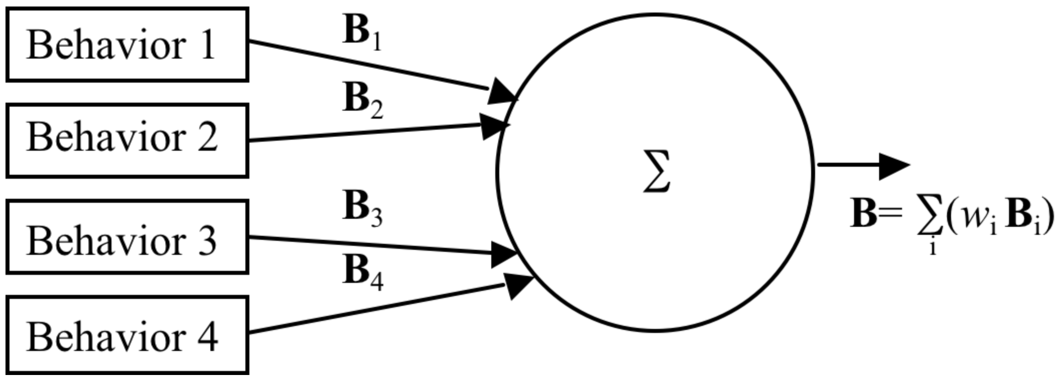
- The figures show the force field **F**, i.e. the gradient of the potential field function
- Each arrow indicates strength and direction of the force
- The force fields are often, somewhat sloppily, denoted potential fields

FIELDS AS BEHAVIORS

Basic idea: Follow the force!

- One field can be thought of as one behavior
- Vector summation to create compound behaviors

FIELDS AS BEHAVIORS

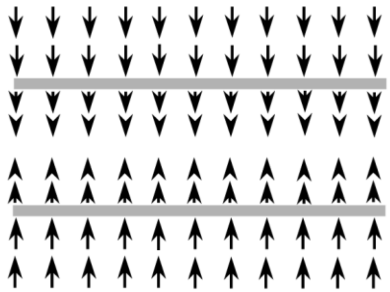


- Four behaviors; each one represented by a potential field, or recursively by other coordinated behaviors
- The output force vectors $\mathbf{B}_1,\ldots,\mathbf{B}_4$ are summed and weighted to produce one compound vector output **B**
- Note: **B** needs to be computed for the robot's location only

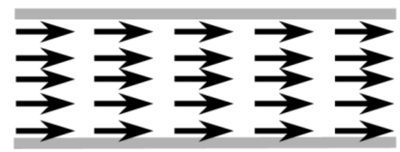
EXAMPLE: FOLLOW-CORRIDOR

constructed as a sum of 3 potential fields:

2 x Perpendicular + Uniform

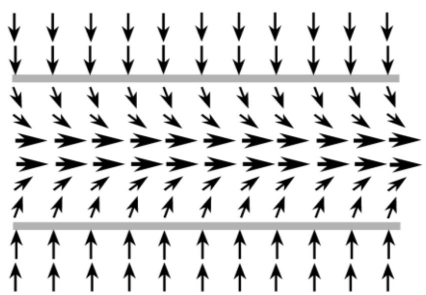


a



b

=



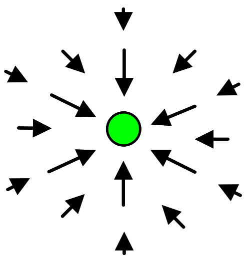
c

The robot “feels” the vector for the current location, then moves accordingly, “feels” the next vector, moves, etc.

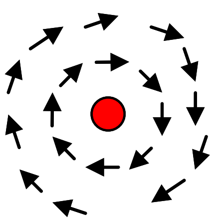
EXAMPLE: DOCKING BEHAVIOR

- Constructed as a sum of 3 potential fields:

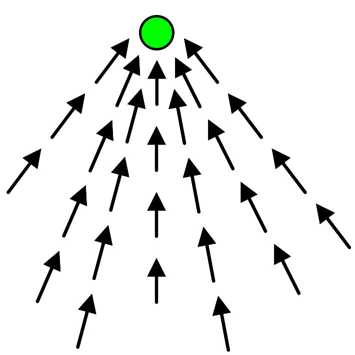
Attraction

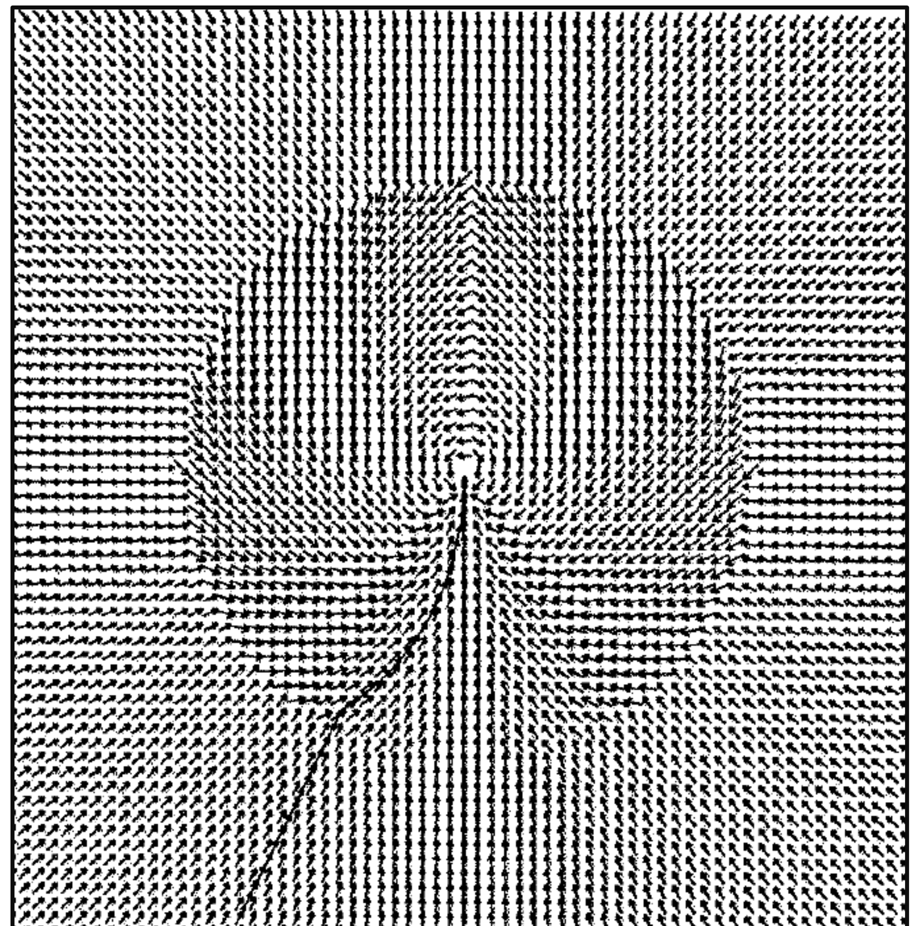


Tangential



Selective attraction



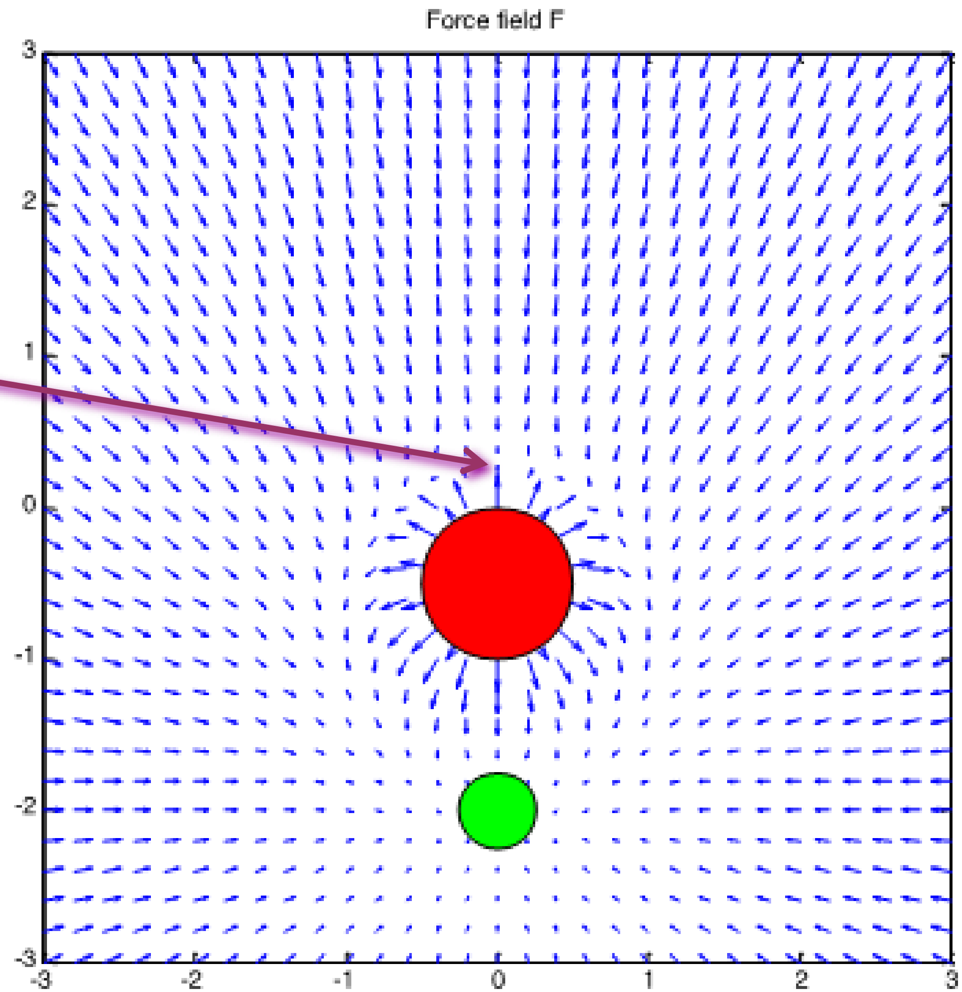


PROBLEM: LOCAL MINIMA

Sometimes the fields add to zero at some points!

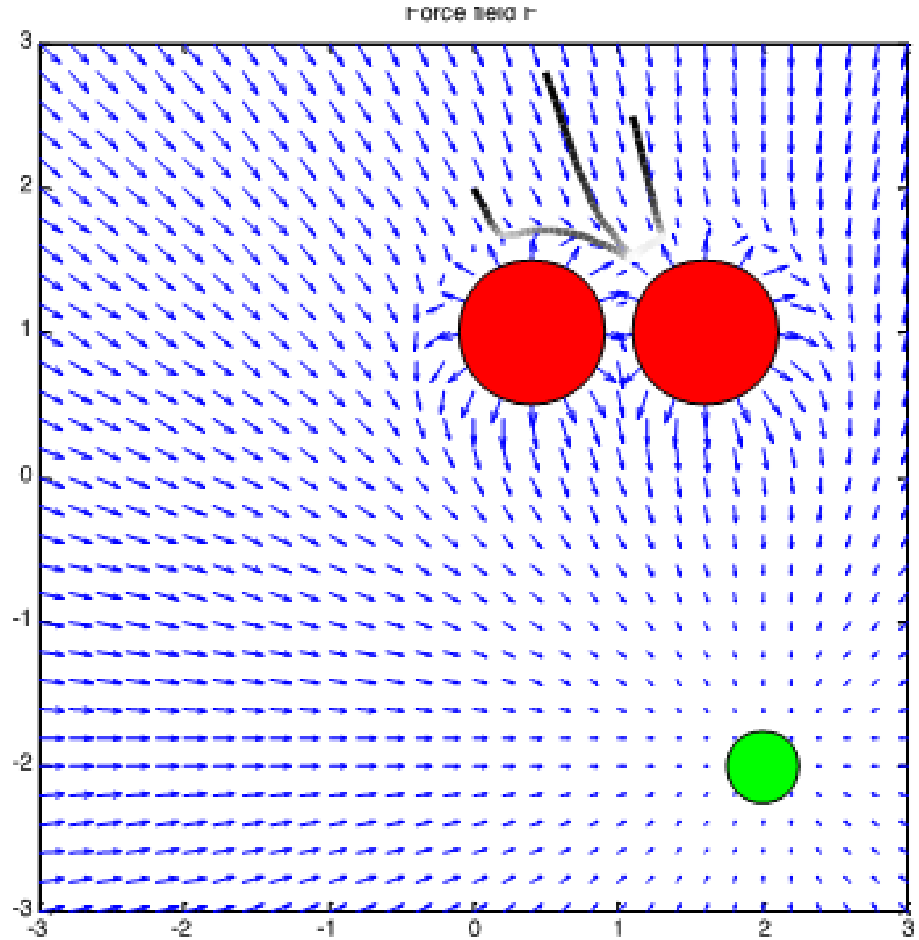
<u>Example:</u>
The repulsive force from the obstacle cancels the attractive force from the goal.

If the robot reaches this point, it will stop!



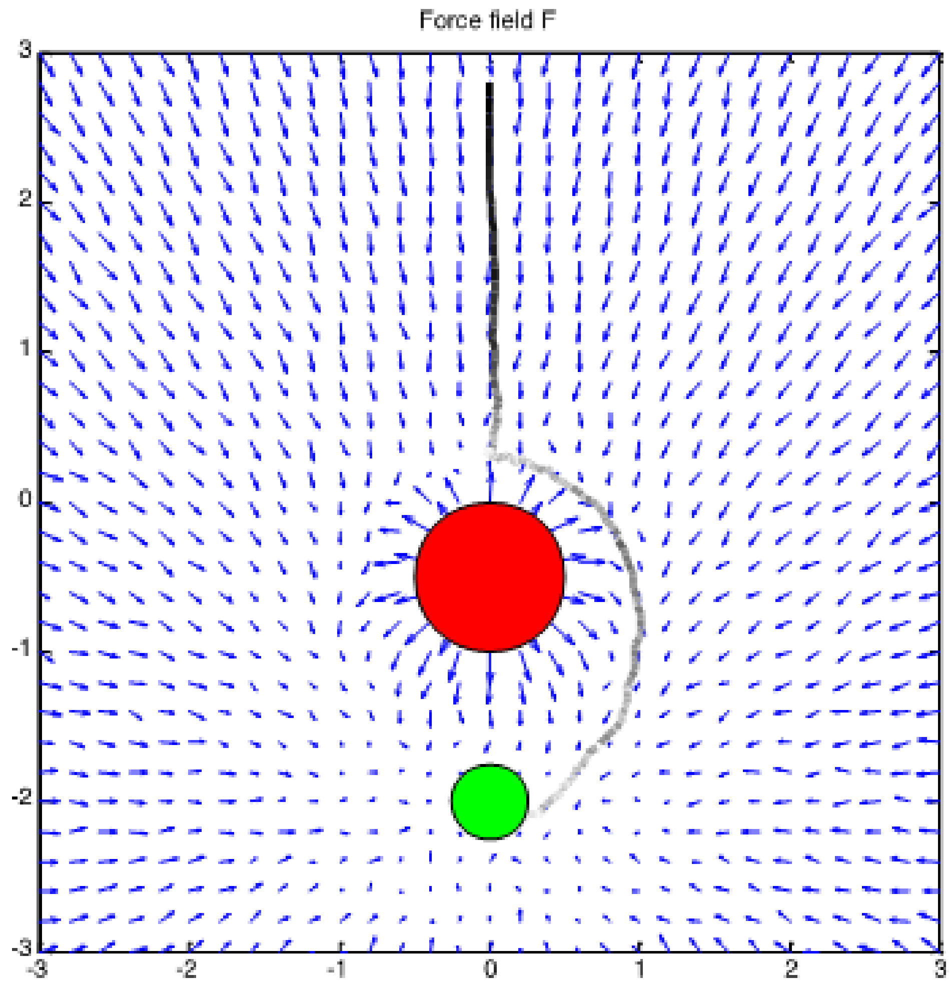
PROBLEM: LOCAL MINIMA

- The robot may be dragged into a local minimum from several different starting positions.
- The gray scale of robot paths corresponds to the speed of the robot.



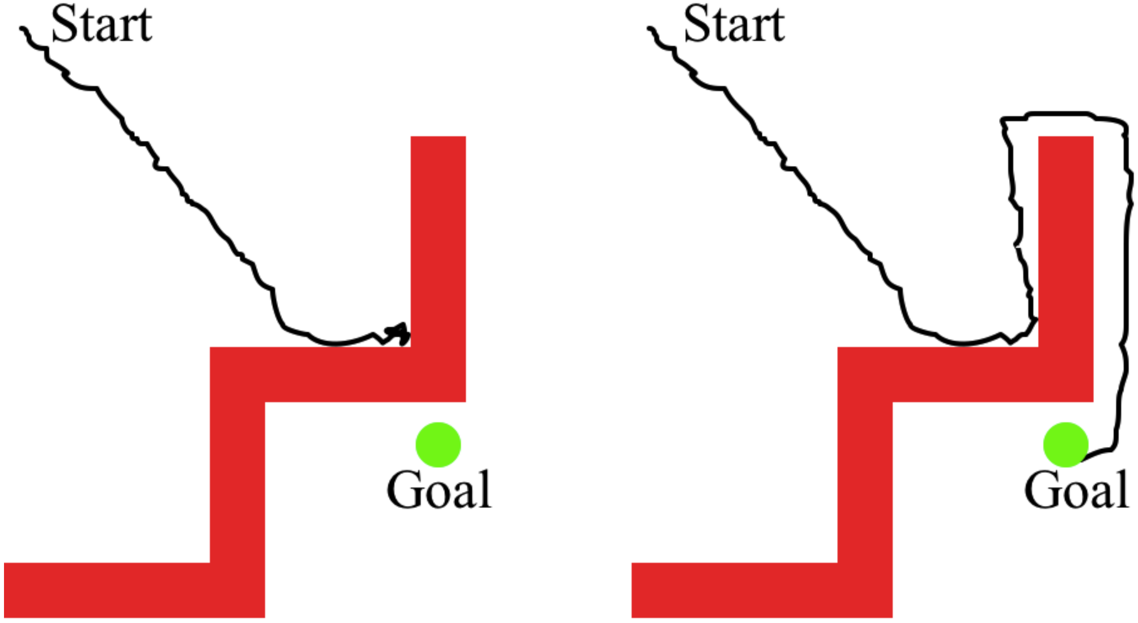
SOLUTION 1: LOCAL MINIMA

- The local minimum is avoided by <u>adding a potential field comprising random values.</u>
- The jagged path is a result of the random force vectors.



SOLUTION 2: LOCAL MINIMA

- Sometimes, adding noise will not solve the problem but rather introduce a **cyclic behavior**: the robot keeps returning to the local minimum
- Solution: Adding a dynamically updated potential field with <u>repulsive forces from all previously visited locations</u>

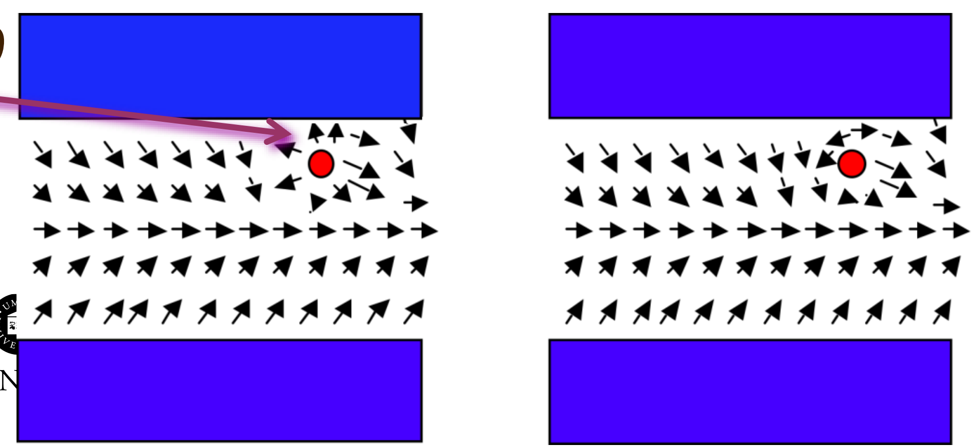


SOLUTION 3: LOCAL MINIMA

- The regular repulsive fields are replaced by a special kind of tangential fields : *Navigation templates*
- The direction of the tangential field is set in real-time not to counter the vector sum of all other fields
- Reduces the risk of attractive and repulsive forces cancelling each other
- Another benefit: Obstacles will be avoided by a force vector that points in the same direction as the general behavior:

The regular repulsive potential field (left) may push the robot into the water

A navigation template (right) avoids this by forcing the robot to avoid the obstacle in a safe direction



PROBLEMS WITH POTENTIAL FIELDS

- Jerky motion (depends on update rates)
- Robot and obstacles treated as points (=> risk of collission)
- The robot is expected to change velocity and direction instantaneously
- Sensitive to local minima

  Three approaches:
  - Injecting randomness
  - Adding an “avoid-past” schema
  - Navigation templates
- Cyclic behavior

  One approach:
  - Adding an “avoid-past” schema that generates a repulsive force from recently visited areas

ADVANTAGES WITH POTENTIAL FIELDS

- Easy to implement and visualize
  - Robot behavior easy to predict by designer
- Support for parallelism
  - Each field is independent of the others and may be implemented as general software, or even hardware, modules
- They can be easily parameterized and configured, also during design or in real-time
- The combination mechanism is flexible and can be tweaked with gains to reflect varying importance of sub-behaviors

COORDINATION OF BEHAVIORS

A general problem in reactive systems is how to handle concurrent and disagreeing behaviors

COORDINATION OF BEHAVIORS

Two major types of coordination mechanisms:

Competetive. “Winner takes all”, “Arbitration”

- Fixed priority methods
  - Subsumption
  - ”Subsumption light”
- Dynamic priority methods
  - Activity based; Select the most “active” behavior.
  - Voting based; each behavior suggests MANY responses

Cooperative. “Behavioral fusion”

- The representation of behaviors must allow fusion.
  - Vector addition such as in potential-fields and schema based approaches
  - Vector addition with dynamic gains
  - Fuzzy logic

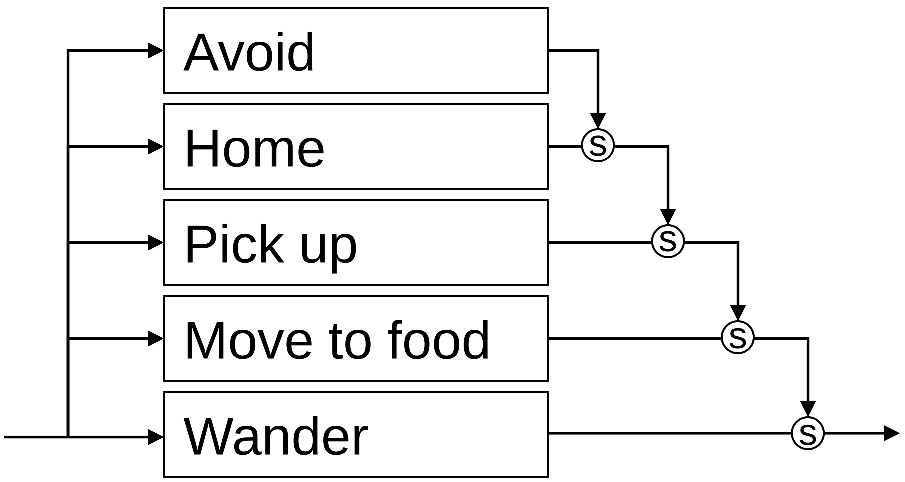
UMEÅ UNIVERSITY

"SUBSUMPTION LIGHT"

Competitive - Fixed priority

Foraging example:

- *Wander*: always active
- *Move to food*: active if the robot sees the food
- *Pick up*: active if food is within reach
- *Home*: active if the robot has food
- *Avoid*: active when an obstacle is in front of the robot



UMEÅ UNIVERSITY

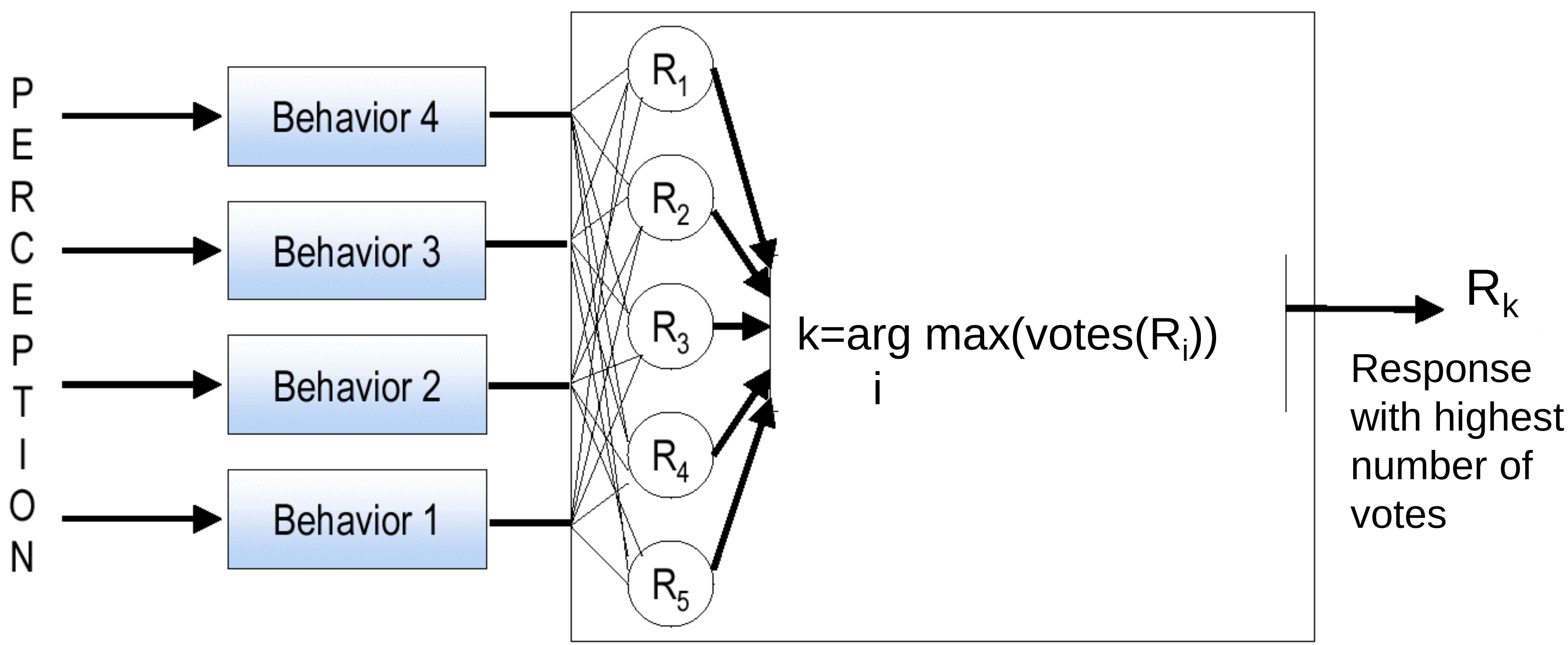
Example from: Srividhya Rajendran, The University of Texas

Voting based arbitration

Competitive - Dynamic priority

- One pre-defined set of discrete Responses for ALL behaviors
- Each behavior votes for each response
- The response with most votes is executed

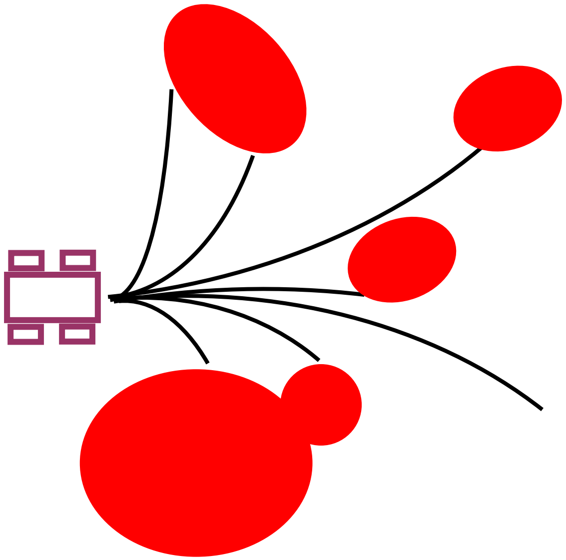
Priorities (no. of votes) may vary during mission

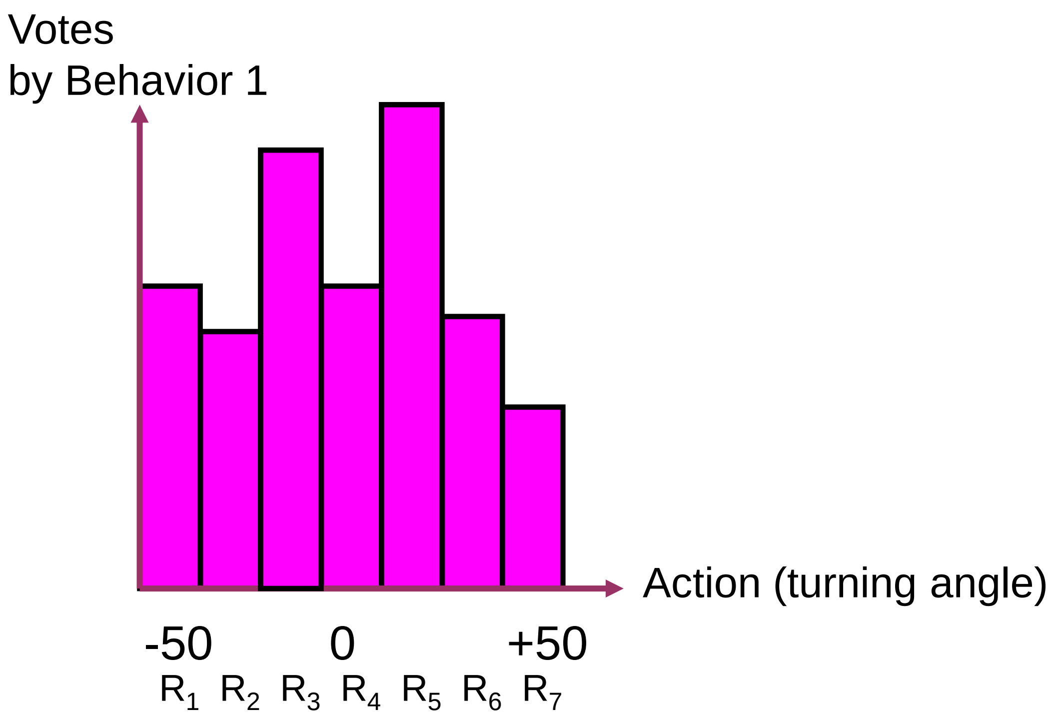


Voting-based coordination

EXAMPLE VOTING

- DAMN – A Distributed Architecture for Mobile Navigation
- Behavior 1 does obstacle avoidance and votes for each response based on estimated time to collision in that direction.
- Behavior 2 does homing and votes for each response based on direction to goal
- Responses R1 to R7 : turning angle

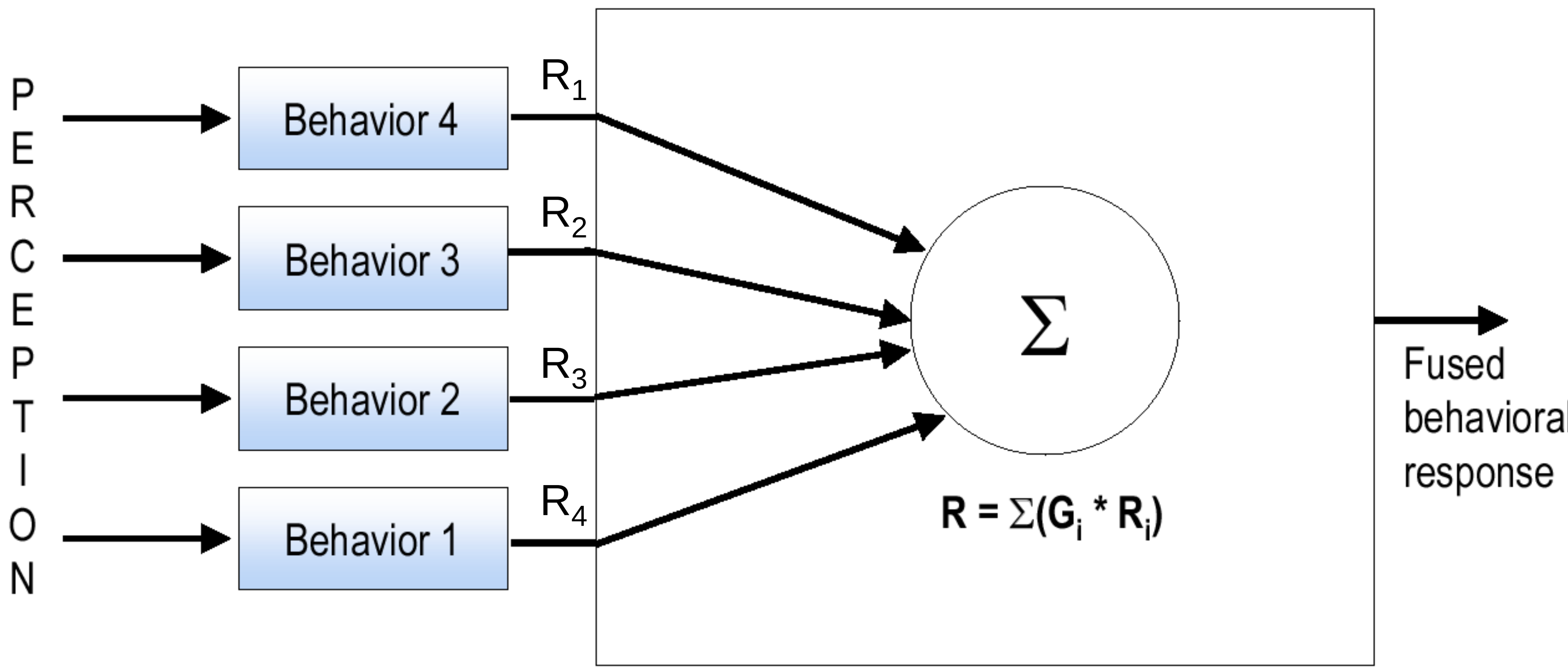




Vector Addition

Cooperative – dynamic gains

The behaviors are fused using gains G_i. Gain levels may vary during mission and depend on the robot's goals and percepts



Behavioral fusion

UMEÅ UNIVERSITY

THE REACTIVE PARADIGM

Advantages:

- Fast reaction time
- Requires less memory and CPU power
- Easy to implement and expand
  - Behaviors are independent
  - No modelling necessary
- Makes no Closed World Assumption
  - Works in dynamic environments that are difficult to characterize and contain a lot of uncertainty

THE REACTIVE PARADIGM

Disadvantages:

- Interacting behaviors can be unpredictable
  "fast, cheap and out of control…" (Rodney Brooks)
- Low level intelligence
  - No memory:
    - Can't handle sequences of behaviors
    - Can't learn anything
  - No world representations:
    - Doesn't know where it is
- Often more art than science

HYBRID DELIBERATIVE/REACTIVE PARADIGM

- Reactivity by the end of the 1980´s was the major trend!
- But not remembering the state of the robot and the world is limiting!
- Question:
  - Should planning be reintroduced? How?
- Answer:
  - Hybrid systems!
  - More on this in the next AI course!


I've been talking to the computer